

FACILITATING MORE FREQUENT UPDATES: TOWARDS EVERGREEN

– A Case Study of an Enterprise Software Vendor's
Response to the Emerging DevOps Trend, Drawing on
Neo-Institutional Theory

Lucas Ersson

Supervisor: Johanna Sefyrin
Examinator: Fredrik Söderström

Abstract

Title: Facilitating more Frequent Updates: Towards Evergreen

Author: Lucas Ersson

Supervisor: Johanna Sefyrin

Background: The last couple of years the trend within the software industry has been to release smaller software updates more frequent, to overcome challenges and increase flexibility, to align with the swiftly changing industry environment. As an effect, we now see companies moving over to capitalizing on subscriptions and incremental releases instead of charging for upgrades. By utilizing neo-institutional theory and Oliver's (1991) strategic response theory, an enterprise systems vendor's response to the emerging DevOps trend can be determined.

Purpose: The purpose of this study is therefore to explore how and why the organization of a Swedish Enterprise Systems vendor, interpret and respond to the isomorphic pressures created by the DevOps paradigm within the software industry environment.

Methodology: This is a qualitative study utilizing the interpretive perspective. The empirical data is collected from ten interviews, all with employees from the case company. The study is carried out with a combination of deductive and inductive approach.

Conclusion: All three types of isomorphic pressures are found to be evident. The coercive and normative isomorphic pressures seem to be exerted to a higher extent than the mimetic isomorphic pressure. Most strategic responses are interpreted as being avoiding or defying in its strategic characteristic, indicating that many initiatives and possible solutions are either ignored or rejected by the organization. From a more holistic perspective, the organization has taken a big step in the right direction. Today, IFS is balancing in the middle of the path towards Evergreen, wanting to evolve the update release concept but also having the mindset of promoting compatibility breaking major upgrade releases.

Keywords: DevOps, Release Frequency, Software Release Management, ERP, Enterprise Systems Industry, Neo-institutional theory, Strategic response framework

Acknowledgements

I would especially would like to thank my supervisor Johanna Sefyrin for the support, without your dedication this study would have been, yet another, empirically heavy case-focused study. I would also like to thank my three supervisors from IFS, Sandra Beskow, Mattias Bladh and Erik Forslund. Lastly, I would like to thank all of the interviewees for contributing to this study with great dedication. Thanks to all of you!

Lucas Ersson, Stockholm 180829

Table of Contents

Abstract	1
Acknowledgements	2
1. Introduction.....	6
1.1 Background.....	6
1.2 Research problem	7
1.2.1 DevOps	7
1.2.2 Neo-Institutional Theory	7
1.3 Purpose and research questions	8
1.4 Delimitations	8
1.5 Target audience.....	9
1.6 Disposition	9
2. Methodology	10
2.1 Model of research design	10
2.1.1 Philosophical assumptions	10
2.1.2 Research model	11
2.1.3 Research approach	11
2.2 Research method	13
2.3 Data collection techniques.....	14
2.3.1 Explorative pre-study	14
2.3.2 Semi-structured interviews	14
2.3.3 Documents	16
2.4 Data analysis approach.....	16
2.5 Research quality and validity.....	17
2.5.1 Triangulation	17
2.5.2 Respondent validation.....	17
2.5.3 Reflexivity	18
2.6 Research ethics.....	18
2.6.1 Honesty.....	18
2.6.2 Informed consent	18
2.6.3 Permission to publish	19
3. Theoretical framework and previous research	20
3.1 Neo-institutional theory	20
3.1.1 Coercive isomorphism.....	20
3.1.2 Normative isomorphism.....	21
3.1.3 Mimetic isomorphism.....	22

3.1.4 Summary.....	22
3.1.5 Analytical framework for strategical response to institutional effects.....	23
3.1.5.1 Acquiescence.....	24
3.1.5.2 Compromise	24
3.1.5.3 Avoidance	25
3.1.5.4 Defiance.....	25
3.1.5.5 Manipulation	26
3.1.5.6 Utilizing the framework.....	27
3.2 DevOps	27
3.2.1 Definition	27
3.2.2 Aims and characteristics.....	28
3.2.2.1 Culture	29
3.2.2.2 Automation.....	29
3.2.2.3 Measurement	29
3.2.2.4 Sharing.....	29
3.2.3 Managing the organizational disconnect and silo mentality	30
3.2.4 Results	31
3.2.4.1 Continuous integration.....	31
3.2.4.2 Continuous delivery.....	31
3.2.4.3 Continuous deployment.....	31
3.2.4.4 Summary of final effects	31
3.3 Software release management	32
3.4 Microservices and monolithic architectures	33
3.5 Software-as-a Service and other software licensing models	33
4. Empirical findings	35
4.1 Introducing IFS – Industrial and Financial Systems	35
4.2 IFS Applications structure.....	36
4.2.1 Phrase definitions.....	38
4.3 Explorative pre-study.....	39
4.3.1 Interviewee Christine	39
4.3.2 Interviewee Alex.....	40
4.3.3 Interviewee Emma	41
4.3.4 Pre-study summary	42
4.4 Main interviews and documents.....	42
4.4.1 Culture and customer focus	44
4.4.2 Decentralization	45

4.4.3	Localizations and local extensions.....	47
4.4.4	Update release frequency	48
4.4.5	Current market trends and IFS Applications licensing models.....	50
4.4.6	Evergreen and the strategic direction for IFS Applications.....	51
4.4.7	Technological architecture	53
4.4.8	Measurement and monitoring.....	54
4.4.9	Automated testing.....	54
4.4.10	Management's role	55
4.5	Empirical summary	56
5.	Analysis.....	60
5.1	Coercive isomorphism	60
5.1.1	Organizational pre-requisites.....	60
5.1.2	The demand for an efficient update release model.....	61
5.1.3	The Global Extension.....	62
5.1.4	The marketing of Evergreen	63
5.1.5	Software licensing models.....	63
5.2	Normative isomorphism.....	63
5.2.1	Organizational pre-requisites.....	64
5.2.2	'IFS Vanilla'	64
5.2.3	Compatibility-breaking major upgrade releases	64
5.2.4	Automation, automated testing and the technological architecture	65
5.2.5	Monitorability.....	65
5.2.6	Strategic responses	65
5.3	Mimetic isomorphism	67
5.3.1	Organizational pre-requisites.....	67
5.3.2	Strategic responses	68
5.4	Summary of analysis.....	69
6.	Conclusions and discussion	72
6.1	Brief introduction to the conclusion and discussion.....	72
6.2	Identifying the presence and extent of isomorphic pressures.....	72
6.3	Interpreting IFS' engagements as strategic responses to the identified isomorphic pressures .	73
6.4	Concluding thoughts.....	74
6.5	Contribution	75
7.	Reflection and future research.....	76
8.	References	77

1. Introduction

This section presents the study to the reader and introduces the subject through a short background and a problematization, which further funnels down into a precise purpose and appropriate research questions.

1.1 Background

The present business environment we experience is fast-paced and ever-changing (Porter and Heppelmann 2014). While some preach that the force of globalization is the main reason (Eisenhardt 2002), others think that the constant technological change with exponential growth constitutes the foundation for the current state of industry markets (Greenberg et al. 2017). However, the common opinion seems to be that both increasing globalization and technological change are present, and that the business environment therefore is changing more rapidly now than ever. In the last few decades, researchers have studied how businesses can succeed in a changing market, and the results imply that to be competitive and capable to capture value, in an atmosphere recognized by unpredictability, an organization's capabilities need to be dynamic (Teece et al. 1997; Katkalo et al. 2010; Eisenhardt 2002).

The exponential technological change continually results in new possibilities and new emerging customer expectations, and therefore also new ways of making profit for businesses. A structured, yet flexible, management of releases will get increasingly important for companies to be able to handle the shifts in market demand for functionality and improvements of software. This management process is known as *software release management*, which Lahtela and Jäntti (2011, p. 10) define as: *"The people, functions, systems and activities needed to plan, package, build, test and deploy software releases effectively into production"*. A release can constitute of both new functional features and patching for bugs (i.e., bug fixes) found in previous versions of the software (Khomh et al. 2012). As a software company, responding to market fluctuations is crucial and the constant change in customer requirements adds further pressure on shorter lead time for software releases to the market (Colomo-Palacios et al. 2017; Claps et al. 2015).

Today we are used to smooth updating processes when it comes to our personal belongings of technological nature (Jain, Mali & Kulkarni 2016). For example, receiving and installing a new iOS version on an iPhone takes almost no time (Mills 2017) and the high quality of each update implies low perceived risks and increases the incentives to take an update (Khoo and Robey 2007). Conversely, people can get confused and annoyed by update messages, and sometimes users even avoid updates, mainly because of unanticipated user interface changes and because they do not acknowledge any problems with the current version (Vaniaa and Rashidi 2016). If the user avoids updating the software, new functionality, bug fixes and security improvements cannot be delivered and a situation where the user would have preferred the update and the software vendor would have preferred to maintain fewer software versions might emerge (Vaniaa and Rashidi 2016). Therefore, assuring that customers enjoy the updating process and the software as expected is crucial to ensure positive perceptions for upcoming updates with the latest refinements and high performance (Mockus and Weiss 2008).

The last couple of years the trend within the software industry has been to break down silo-walls between development and operations-related divisions, and de-compartmentalize organizations, mainly development and operations-related divisions, to ultimately enable releasing smaller software updates more frequently, to overcome challenges and increase flexibility (Fitzgerald and Stol 2014), thus, aligning to the swiftly changing industry environment (Porter and Heppelmann 2014). This trend is referred to as an organizational paradigm within software development, called *DevOps* by recent literature (Kamuto and Langerman 2017). As an effect of DevOps, we now see companies moving over to capitalizing on subscriptions and incremental releases instead of charging

for upgrades (Caray and Macaulay 2018). For example, Microsoft now offers their Office software package as a Software as-a-Service (SaaS) and they have announced that Windows 10 will be the final version and future functionality will instead be delivered in incremental update releases (Microsoft 2018).

1.2 Research problem

This section describes the perspective of DevOps as an emerging organizational paradigm, and then briefly connects it to the empirical prerequisites of the studied case company. Thereafter the neo-institutional theory is presented and the research problem is formed by combining DevOps as a concept with neo-institutional theory.

1.2.1 DevOps

The trend to deliver incremental and more frequent software releases is ascribed to a new upcoming paradigm of software development, referred to as *DevOps* by the literature (Kamuto and Langerman 2017). DevOps can be described as a concept which emphasizes collaboration and eliminates the split and distance between the developing and operational functions within an organization, with the main promise to enable faster and more frequent software releases (Wettinger et al. 2014; Humble and Molesky 2011). In the last couple of decades, the focus of research and practice has mainly been on improving the agile software developing methodology, an approach to software development that utilizes short development cycles in order to accomplish value and improvement in incremental steps (Khom et al. 2012). During this time, less attention has been given to the processes of software release management, e.g., planning, deployment, operations and maintenance (Fitzgerald and Stol 2014). Now, the paradigm is shifting and focus is being put on DevOps, which is often seen as a cultural movement in combination with practices that enable a more efficient software release management (Smeds et al. 2015).

To successfully compete in the fast-paced software market, ultimately, offering close-to immediate deployment of software code to customers as soon as it is developed is needed (Chen 2015). This can be achieved by enabling *Continuous Integration*, *Continuous Delivery* and *Continuous Deployment* which is a long-term goal for many software companies (Claps et al. 2015). As a step towards reaching this goal, companies are now speeding up their deliveries to become more dynamic while offering their new innovations and refinements more quickly to their customers (Khom et al. 2012).

When the R&D division of a Swedish Enterprise Systems vendor presented a proposal for increasing the release frequency of their ERP software, it was declined by the regional consultancy and support divisions. This study focuses on the foundation for the declination and further investigates in the vendor's responses to the current market trend.

1.2.2 Neo-Institutional Theory

Neo-institutional theory builds upon the institutional theory that is based on the belief that organizations are social constructions that adapt and change in reaction to internal and external forces (Hu et al. 2007). Neo-institutionalism extends the idea by stating that similarly positioned organizations acting in the same industry, will be shaped by phenomena in their institutional environment and over time, become isomorphic with it, i.e., they are influenced by institutional isomorphism (Suddaby 2010, Hu et al. 2007). Hu et al. (2007) confirms that neo-institutional theory provides an appropriate lens to explain how the institutional environment influence the decision making of organizational actors and therefore also the behavior of the organizations themselves.

Within information systems research, relatively few studies explore emerging organizational phenomena, such as DevOps, with an institutional theoretical lens (Su 2011). One example is Gazza and Wahid (2015) who draw upon Oliver's (1991) institutional strategic response framework when

analyzing Norwegian public-sector organizations' adoption of cloud-based software from a neo-institutional perspective. Within DevOps literature, most studies either focus on the challenges organizations run into during the adoption phase (Kamuto and Langerman 2017; Neely and Scott 2013; Colomo-Palacios et al. 2017; Farroha and Farroha 2014), or review the current definitions for the subject (Dyck et al. 2015; Smeds et al. 2015; Lwakatare 2016). Because of the absence of literature covering the emerging DevOps paradigm from a neo-institutional perspective, this study will focus on unpacking DevOps as an emerging organizational paradigm, and then analyze an enterprise systems company's response to it.

1.3 Purpose and research questions

The research study at hand is executed as a case study, combining a real business assignment with research. This means that the study must satisfy multiple stakeholders with different perspectives and perceptions. By taking a holistic perspective and consolidating the interests of both the field of research within information systems and management and the studied organization, this study aims to produce valuable results to both parties. By uncovering the evolvement of software release management processes accompanied by other related subjects within the vendor's organization, and utilizing the framework provided by Oliver (1991) and later updated by Gazza and Wahid (2013), an analysis of the vendor's response to DevOps from a neo-institutional perspective will be performed.

The study aims to result in an extensive report of the development and operations processes of the Swedish Enterprise Systems vendor's ERP software, which might be applicable in the vendor's work for a more efficient software release management. Regarding the research frontier, the study should contribute by filling a previously untouched gap where a Swedish Enterprise Systems vendor's approach and response to DevOps is analyzed from a neo-institutional perspective.

The purpose of this study is therefore to explore how and why the organization of a Swedish Enterprise Systems vendor, interpret and respond to the isomorphic pressures created by the DevOps paradigm within the software industry environment.

The purpose funnels down into the following research questions:

- I. What isomorphic effects can be identified as exerting pressure onto a Swedish Enterprise Systems vendor, and to what extent are they evident?
- II. How can the Swedish Enterprise Systems vendor's engagements within DevOps be interpreted as strategic responses to the identified isomorphic pressures?

1.4 Delimitations

The main delimitations of this case study are hereby presented. Firstly, within the vendor's organization there is a handful of various business software. This case study only focuses on the release management processes regarding the vendor's main ERP application. Therefore, the vendor's approach to the underlying bottlenecks of the software release management, and their responses to the isomorphic pressures of the software industry, is only determined concerning their ERP software. Secondly, during the introduction phase of the study the whole organization of the vendor is looked at, but after the pre-study mainly the Scandinavian region is covered by the research. This delimiting has its roots in keeping the research within reasonable proportions. Further studies of the other geographical regions of the organization is therefore proposed. The theoretical framework is delimited in regards to relevance. I.e., Oliver's (1991) strategic response framework is used, but not the predictive dimensions of the strategic responses because of the limited relevance to the empirical case. Another example is that DevOps theory is used, but since the vendor is not utilizing continuous deployment, low focus is being put on that dimension of the DevOps theory.

1.5 Target audience

The subject of the study was determined in discussion with representatives from the vendor, and has been continually shaped during the progress.

The study mainly has three groups of target audience; the vendor's employees of various hierarchical levels, students, or other people, interested in information systems research, organizational management, DevOps and software release management. Lastly, the academic field of neo-institutional theory is treated as the third target group of the thesis study. In order to accommodate the interests of all three target groups, language and structure is set to appeal people from diverse backgrounds.

1.6 Disposition

Firstly, the Introduction chapter presents the background and the research problem briefly, and funnels down into a purpose and the research questions. Then, the Methodology chapter covers the model of research design, research method, data collection techniques, data analysis approach, quality and validity and research ethics. The Theoretical framework chapter presents the theoretical concepts and previous research used as a theoretical spine throughout the study. The Empirical findings chapter then covers the information collected at the vendor's premises, including the results of the multiple interviews conducted. Then, the Analysis chapter connects the empirical findings with the theoretical spine to lay the analytical foundation for the Conclusions and discussion chapter that follows. Then the Reflection and future research chapter discusses reflectively on the thesis work and proposes related future research.

2. Methodology

This chapter describes how the study is performed by presenting and motivating the choices that constitute the overall structure of research methodology. First the model of research design is presented, followed by the research method, the data collection techniques and the data analysis approach. Finally, the research quality and ethics are assessed.

2.1 Model of research design

While Justesen and Mik-Meyer (2013) think of the philosophical assumptions as the first decision that should take place when preparing a study, Myers (2013) instead proposes that the research method, e.g., qualitative or quantitative, should be decided first. In my opinion it is more rational to first pick the philosophical perspective which lays the foundation for choosing the research model and the research approach. The process is illustrated in Figure 1, and further described under the three subsequent headings.

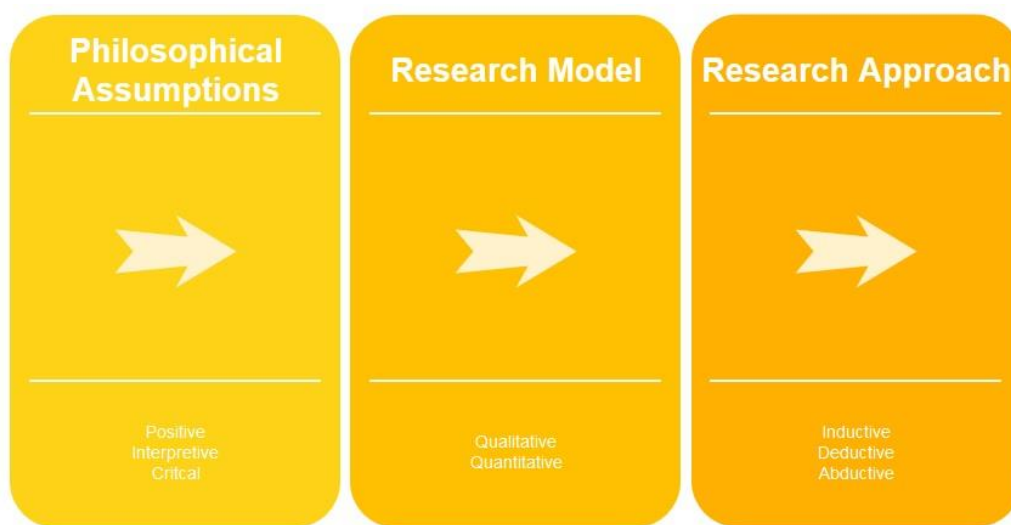


Figure 1: Model of research design (self-produced).

2.1.1 Philosophical assumptions

Since an organization's responses to isomorphic pressures are sourced from the decision making and acting of the people within the organization (Hu et al. 2007), the researched phenomena would not be present without the affecting presence of humans. The hermeneutic philosophical epistemology is based on the belief that the reality is created and shaped by the interaction between humans and the products we create and use, and since the research study is based upon interpreting an organization and the peoples' views within it, it is determined as a good fit (Justesen and Mik-Meyer 2013). The contrary alternative would have been positivism, which relies on the assumptions that reality is an objectively created matter which content in general can be measured (Meyer 2013). Because of the measurability, positivist studies are often set up to test theories to further increase the predictability of similar matters, i.e., contribute to generalization (Myers 2013).

In general, the choice of philosophical perspective is determined by the relation to the underlying epistemology of the research (Myers 1997). There are two main philosophical perspectives related to the hermeneutical epistemology; interpretive and critical (Myers 2013; Justesen & Mik-Meyer 2013). A researcher with an interpretive philosophical perspective focuses on accessing reality through socially constructed concepts, such as language or consciousness, to reach a holistic understanding of each party's perception of a situation (Myers 2013; Justesen & Mik-Meyer 2013). Instead, a

researcher exercising the critical perspective utilize the same contextual reality access to illuminate the restrictions created by the status quo in a social context (Myers 2013).

Choosing between the two perspectives is a challenge, especially since the purpose of the study is to interpret and analyze the subjective experiences of employees within the vendor's organization, and at the same time illuminate overlooked limitations in the release management process. The literature suggests that the distinctions between the two perspectives are not that clearly cut, which enables the researcher to utilize a combination of them when conducting a study with focus on the hermeneutic epistemological assumptions (Myers 1997; Justesen & Mik-Meyer 2013).

Therefore, I've decided to mainly utilize the interpretive philosophical perspective, with a touch of its critical sibling. By interpreting the subjective experiences of employees and creating a holistic understanding of the researched phenomena, I expect to disclose challenges and opportunities related to the software release management processes in a larger context, thus being able to, in a critical manner, illuminate overlooked bottlenecks and hindrances.

Choosing the interpretive perspective (with a touch of critical) will have affections for the general methodological alignment of the study, which will be presented in the forthcoming segments.

2.1.2 Research model

According to Kaplan and Maxwell (2005) the primary goal of a qualitative research model is to reach understanding by investigating the perspectives and the context of the people involved in the studied situation. Their thoughts align with my choice of an interpretive philosophical perspective, which focuses on interpreting peoples' different views and understanding the context of them. Myers (1997) even advocate that this concept of interpretive understandability is lost when the data is quantified. Hence, a quantitative model seems inappropriate for the kind of subject I am studying.

However, a qualitative research model is suited for unpacking different views within a community of people, with the primary strength being the ability to examine underlying values and assumptions to understand what drives an organization in a specific direction (Choy 2014). Kaplan and Maxwell (2005) argue that a qualitative research model is used to investigate process dynamics rather than statics, which is better examined with a quantitative research model. Similarly, they also say qualitative research models can be utilized for understanding a particular case, rather than generalizing to a larger set (Kaplan and Maxwell 2005).

Kaplan and Maxwell (2005) conclude by stating two scenarios where qualitative research models are more effective and better suited than qualitative equivalents:

- In a situation where the researcher needs to understand how, and why, people think and feel the way they do, what their perspectives are and how they affect what is happening
- When studying contemporary emerging processes, rather than results that have already taken place

The two points above align perfectly with the focus of this research since I, as a researcher, need to adopt a holistic understanding for the different parties of the organization and its release management processes. Additionally, responding to the isomorphic pressures of the software industry is a contemporary process within the organization. It is therefore evident that this study needs to incorporate a qualitative research model.

2.1.3 Research approach

Kaplan and Maxwell (2005) argue that the research can take or combine two different approaches:

- Deductive approach
- Inductive approach
- Abductive approach

With the deductive approach, the research begins with establishing a theoretical framework, constituting of previous studies and theoretical models, which in turn frames and sets guidelines for the empirical data collection (Kaplan and Maxwell 2005). The inductive approach, instead focuses on avoiding prior hypotheses or theoretical constructs when collecting empirical data, and is more explorative in its behavior (Kaplan and Maxwell 2005). The combination of the two is called abductive approach, which focuses on combining both the explorative and framing characteristics of the others (Dubios 2002). According to Eisenhardt (1989) it is important to approach a case study with as near a clean theoretical slate as possible, as it may bias and limit the findings. Kaplan and Maxwell (2005) however clarify that the inductive approach does not try to eliminate the researcher's previous theoretical bent, rather, it tries to have the researcher work outside the narrowing effect of theory, which the deductive approach has the researcher work within.

The determined philosophical assumptions indicate what approach is the most sufficient (Justesen and Mik-Meyer 2013). For example, a positivist study often incorporates setting up a hypothesis based on theory to test in practice, while an interpretive study is more likely to be of an explorative kind and therefore progress by interpreting and observing patterns which are then connected to theory (Myers 2013).

In this research case, interpreting and creating patterns of many individual employees' subjective experiences is mandatory to understand the challenges and opportunities from a holistic perspective. Simultaneously, the framing of the research problem is constituted by the neo-institutional theory, hence why I have chosen to conduct the study utilizing a combination of the deductive and inductive approaches, in other words an abductive approach. The abductive approach used in this particular study is based on an introducing deductive part that is relatively straight forward, and a later occurring inductive part that comes with the drawback of being a very time-consuming procedure (Myers 2013). To ensure I do not get overwhelmed and led astray by the explorative sense of the inductive part of the research approach, it will be utilized in an iterative manner throughout the thesis process in order to continually align empirical data to the theory of the deductive approach, as seen in Figure 2.

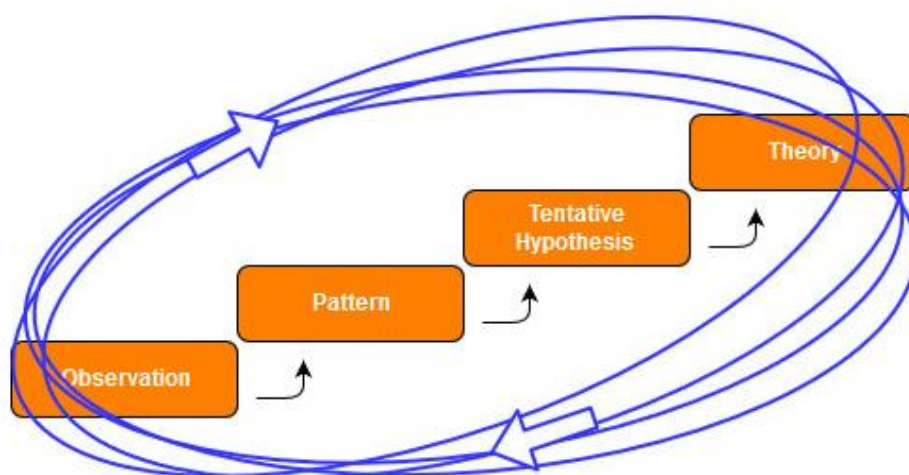


Figure 2: Iterative inductive process of the abductive approach (self-produced).

The methodological result will be a research study conducted off a neo-institutional theoretical bent, with the literature exploration being based on the continually collected empirical patterns discovered through observations and interviews.

2.2 Research method

This study is set to be conducted as a case study of the vendor's organization at their premises. Myers (1997) states that a research method is strategy of inquiry for the collection of empirical data, based on the underlying philosophical assumptions determined for the study. For an interpretive research, the case study method is aligned well. Mainly, because a case study research is defined by focusing on answering questions of the "How?" and "Why?" nature (Myers 2013).

According to Myers (1997) the case study form of research method is the most commonly used method among information systems research, and is especially applicable when exploring a phenomenon in the context of a real-life situation (Myers 2013). Myers (2013) describes a case study research method as *the vehicle* when conducting an explorative research of a new subject area. When conducting a research within a recent and contemporary subject, like DevOps, the explorative interpretive format of a case study research method therefore seems to be a good fit.

Yin (2003) describes a case study as:

- Investigating a contemporary phenomenon within its real-life context.
- Where the boundaries between phenomenon and context are not clearly evident.
- Coping with the technically distinctive situation in which there will be many more variables of interest than data points.
- Relying on multiple sources of evidence, with data needing to converge in a triangulating fashion.

The performed study is affected by all four of Yin's descriptions. First, the investigated phenomenon is contemporary and of great interest for the organization. According to Myers (2013) it increases the chance of the researcher receiving support from the organization. Secondly, the boundaries between the phenomenon and the context of this specific case study are not clear, but the well formulated and distinctive research questions help clarifying the divide. To cope with the third point, I will follow Myers (2013) recommendation regarding utilizing interviewees with diverse perspectives to constitute a foundation for a holistic understanding. Also, identifying key individuals with decision making power and knowledge is valuable for acquiring a great depth in the empirical data collection (Myers 2013). Therefore, I have chosen to execute the interviews starting with interviewees inhabiting diverse perspectives, while finishing with key interviewees for each subtopic according to the structure presented in Figure 3. This structuring of interviewees will hopefully let me habit a holistic perspective before entering the interviews with the key interviewees, maximizing the quality of the empirical data collection.

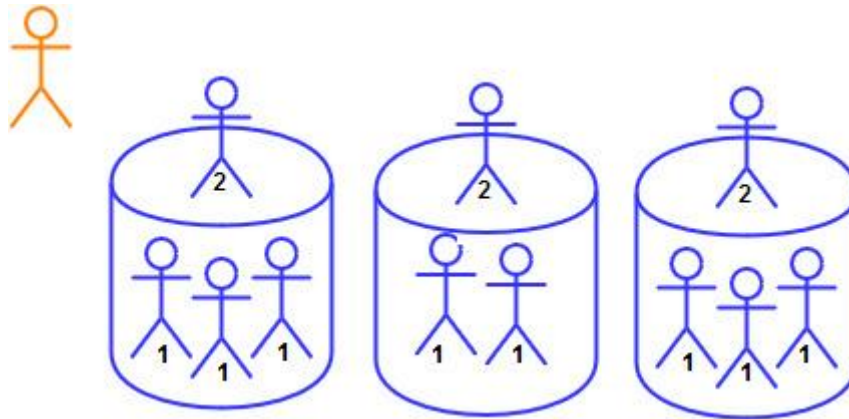


Figure 3: Structure of interviewees (self-produced).

I intend to meet Yin's fourth requirement regarding the case study needing to be heavily reliant on utilizing multiple data sources, i.e., triangulation. By utilizing the internal documentation available within the organization, I create an underlying foundation of knowledge preemptively for the interviews.

Conducting this research as a case study incorporates doing the actual research work at the vendor's premises, which will yield additional positive effects beyond from being given access to internal documentation. Mainly, in the form of the researcher learning the "organizational language", thus understanding the views of the interviewees better. This aligns with Myers (1997) requirement for an interpretive researcher to explore a social or cultural phenomenon from the inside rather than from the outside.

2.3 Data collection techniques

In this topic, the techniques for collecting empirical data are presented. I have utilized an explorative pre-study in order to aim the scope of the study accordingly, before conducting the main interviews. Lastly documents are used to confirm and strengthen the weak parts of the result from the main interviews.

2.3.1 Explorative pre-study

As an initiation of the study, an explorative pre-study was performed to set an approximate scope for the study. Three semi-structured interviews were conducted in an early stage to investigate and determine what parts of the software release management are the most challenging and why. The pre-study helped in delimiting the practical studies, and adopting a holistic mindset over the organization. Apart from narrowing down the scope, the interviews within the pre-study produced leads to a handful of relevant interviewees within the organization as well as they gave me preparation in the form of interview training before the extensive collection of empirical information that will be conducted at a later stage in the study.

2.3.2 Semi-structured interviews

Within qualitative research, data collection techniques are utilized to collect empirical data in the form of words or acting (Kaplan and Maxwell 2005). Within case studies, interviews and documents are the most commonly used techniques, with participant observation being used more seldom (Justesen and Mik-Meyer 2013; Myers 1997). Myers (2013) refers to interviews as a pair of night goggles; they permit us to examine what is ordinarily viewed at, but seldom seen. I find Myers' (2013) view of interviews as a data collection technique very intriguing. Especially in the sense of a tool that can be used to unpack and uncover diverse sets of perceptions, since I, within this study,

intend to gather and consolidate those into a holistic understanding of a previously concealed phenomenon.

An interview can be mainly classified in three different ways depending on its adherence to predetermined questions and time frame:

- Structured
- Semi-structured
- Unstructured

According to Myers (2013), a structured interview has predetermined questions, seeking strictly delimited answers, often within a narrow time frame. A semi-structured interview has some predetermined questions formed as a guide for the interviewer (Justesen and Mik-Meyer 2013; Myers 2013). The guide does not need to be followed it strictly, rather new questions are encouraged to emerge during the interview, and usually the time frame is more loose compared to a structured interview. (Justesen and Mik-Meyer 2013; Myers 2013). An unstructured interview is very loose, both regarding adherence to pre-formulated questions and time frame. In this case, pre-formulated questions are often very open or non-existing, and the time is seldom limited (Justesen and Mik-Meyer 2013; Myers 2013).

A semi-structured interview contains the best from both structured, and unstructured interviews. It contains the structure and guiding characteristics of a structured interview, at the same time as it lets the interviewee talk freely, and touch other interesting subjects which will add to the important holistic view of the interpretive perspective (Kaplan and Maxwell 2005; Justesen and Mik-Meyer 2013). I've chosen to conduct the interviews for this case study utilizing a semi-structured form. It is the form of interview I've utilized the most in previous case studies from earlier experiences, hence it is also the form I am the most comfortable with. According to Myers (2013), the more comfortable the researcher is with an interview technique, the more valuable it is going to be for the research. He also states that the choice of interview technique depends on the research method and the philosophical assumptions used (Myers 2013). Since I utilize an interpretive perspective for a case study, with a mostly inductive, i.e., explorative, approach, the semi-structured interview is determined as a proper fit (Justesen and Mik-Meyer 2013). I further relate to Choy (2014) who explains that a semi-structured interview technique allows the interviewees to raise the issues that matter the most to them, and therefore also to the explorative research.

According to Myers (2013), the major advantage of the semi-structured interview form is that the interviewee is allowed to talk freely. On the contrary, it is also seen as the major disadvantage since the interviewee might not be talkative enough, or too talkative which then leaves the interviewer with heaps of irrelevant data (Myers 2013). In general, the more comfortable the interviewee is in the interview situation, the more valuable will the disclosure of information be (Myers 2013). In order to keep the interviewee in a comfortable mood, I will follow Justesen and Mik-Meyer's (2013) recommendation regarding the importance as an interviewer to reflect over how your own acting, age and experience level and acknowledge how it may affect the situation (Justesen and Mik-Meyer 2013). For example, I let the interviewees decide the scheduling for the interviews and ensure to interview people on their respective floors in the vendor's building. By utilizing the technique of mirroring (Myers 2013), I will re-use words and phrases the interviewees use, when asking subsequent questions or making comments. By using the interviewees' own language in this way, I hope to elicit the innermost and correct views of his or her own experiences.

2.3.3 Documents

According to literature, utilizing documents as empirical information can be very valuable when conducting a case study (Myers 2013; Justesen and Mik-Meyer 2013). Mainly, because they reflect the culture of the studied organization (Myers 2013), but also since they provide information of how they are used by the employees, thus offering an angle of the subjective reality the employees take part in (Justesen and Mik-Meyer 2013). In my opinion, the latter should be valuable when conducting a case study from the interpretive perspective, where the subjective reality of the employees is a distinct focus.

Internal documents also have the advantage of providing information or guidelines that interviewees can have difficulties remembering (Myers 2013). Additionally, this type of information can be helpful when framing interview questions (Justesen and Mik-Meyer 2013). However, a pre-requisite for collecting information from internal documents is getting access to them by the organization (Myers 2013).

The organization show extensive trust in me and for the research I conduct, hence why I have access to strategy documents and other relevant internal communication, such as meeting recordings and internal surveys. The most important documents used are strategic documents stating R&D Targets for 2016, 2017 and 2018, various press releases and a video communication where the newly appointed CEO of the vendor discusses his thoughts on the organization and its future. These documents will be used in twofold, partly to constitute the foundation for my own understanding of the complex organization, but also as an empirical interconnection between the subjective realities of the interviewees, which will be needed in order to recognize a holistic perspective of the organizational interaction and the software release management.

2.4 Data analysis approach

In this study, the data analysis approach is based on the hermeneutic mode of analysis (Myers 2013). This mode of analysis with the same name as the philosophical approach mainly utilizes the concept of the hermeneutic circle which refers to interpreting individual parts of a text or text-analogue for the researcher to understand the full meaning of it (Myers 2013). To succeed with the analysis and 'close' the hermeneutic circle of interpretation, Myers (2013) says that the understanding must repeatedly move between individual parts of the text, to the whole of it.

The phase of data analysis can be difficult to separate from other phases when conducting a qualitative study (Myers 2013), especially compared to the quantitative research model where the analysis phase is clearly distinguished from the others (Myers 1997). In this study, the analysis started the same moment I conducted the empirical pre-study, and compared the results to my previous experiences. It aligns well with how Kaplan and Maxwell (2005) see data analysis within a qualitative study as an ongoing activity that intertwines with both the data collection and the research design. Myers (1997) agrees, and even states that it is probably more accurate to refer to different modes of analysis throughout the study, instead of a single phase of analysis. However, I will just refer to data analysis as a single ongoing process, since it is an iterative interpretive process that is present in the background continually throughout the study.

Interpretation is a key factor within the data analysis activity for two reasons. Firstly, since a qualitative researcher almost always ends up with a chunk of data that has to be reduced, interpretation is used to help focus on the most relevant aspects of the data (Myers 2013). Secondly, interpretation is needed to make sense of the whole, by understanding different parts and their interrelationships (Kaplan and Maxwell 2005). Combining these two types of interpretation is very effective in a situation where there are cloudy, incomplete and contradictory views of organizational processes and events (Myers 2013). Since the purpose of this study focuses on making sense of

different views and events within the large organization of the vendor, extensive focus will be put on analysis and interpretation. For example, transcribing and coding of the interviews is done carefully to let the analysis process begin early and foster an emerging holistic view of the complex organizational string of events related to the software release management of the vendor's software application. The coding of the interview transcripts is done properly by coloring important aspects in different colors, based on their respective empirical subtopic. In this way, I ensure a distinct structuring of the empirical data, while at the same time beginning the analysis process early.

The structuring of the analysis chapter is set based on the theoretical framework. The subtopics are entitled after the three main types of isomorphic pressures, under which the distinctness and extent of each is discussed, as well as the organization's reaction to them. By structuring the analysis chapter in this way, I hope to distinct the structure of the analysis chapter from the empirical chapter in a reasonable manner.

Myers (2013) mentions that the most significant difficulty with extensive interpreting is knowing when to stop. I think he means that the recursive process of interpretation is, in sense, infinite. To counter this challenge, I will ensure to keep communicating with my supervisors, both at organization and the university to continually evaluate the state of the analysis.

2.5 Research quality and validity

Assessing quality within qualitative research is not as straight forward as within quantitative research, where focus is being put on generalizability, validity and reliability (Mays and Pope 2000). As an effect, it has been debated what criteria that are appropriate and should be used for assessing quality within qualitative research (Justesen and Mik-Meyer 2013). Consensus is that by ensuring validity of the data collection techniques, high research quality will be sustained (Mays and Pope 2000; Myers 2013; Justesen and Mik-Meyer 2013). Therefore, this section will further discuss validity as a criterion for assessing quality, and how it is assessed to reach quality within this case study research.

Justesen and Mik-Meyer (2013) treats validity as a criterion of quality that assesses the extent to which the data collection techniques reflect the researched phenomena and variables that are of interest for the researcher, and their contribution towards answering the research questions. This section discusses three different parts of the validity criterion; triangulation, respondent validation, and reflexivity.

2.5.1 Triangulation

Triangulation is often seen as the most suitable method to test the validity of research, mainly because of its simple, yet effective, approach (Mays and Pope 2000). Within a research study, triangulation means strengthen the findings by cross-validating them against other data sources or collection methods (Kaplan and Maxwell 2005). Mays and Pope (2000) explain it further, and state that triangulation can be achieved by comparing results between interviews, observation and documents, or for example by diversifying interviewees. In short, the concept of triangulation fosters increased research credibility by letting different kinds and sources of data to converge (Kaplan and Maxwell 2005). To reach increased credibility in this study, triangulation will be met in two ways. Partly, by utilizing both interviews and documents as data collection techniques, but also through diversifying interviewees, ensuring different perspectives and perceptions, as explained before.

2.5.2 Respondent validation

According to Kaplan and Maxwell (2005) respondent validation is the most important method to ensure correct interpretation of the participants' contribution to the study. The case study aims to

identify and interpret diverse subjective experiences regarding how the vendor is approaching the bottlenecks for an increased release frequency of the vendor's software application. In order to ensure validity of the identification and interpretation processes, the empirical findings of each interview are verified with my supervisors at the organization. This should be an appropriate respondent validation method since both Kaplan and Maxwell (2005) and Justesen and Mik-Meyer (2013) claims that it should be just as effective to involve other relevant people with familiarity to the subject in the respondent validation. Mays and Pope (2000) advocate that it might even be an advantage, since the researcher's condensation of the diverse perspectives in general is designed for a wider audience and therefore might differ from the anticipation of the participants.

2.5.3 Reflexivity

Reflexivity is a criterion of validity incorporating a process of reflection for the researcher, where he or she should evaluate how biases such as previous experiences, interests or perceptions have shaped the research results (Mays and Pope 2000). Kaplan and Maxwell (2005) illuminates that both the researcher and the research process itself can have affecting properties for the collection and analyzation of the data, and therefore play a role in the study. Justesen and Mik-Meyer (2013) add that the reflections need to be transparent to the reader in order for the study to attain reflexivity. Therefore, all my reflections and opinions that comes apparent during this research study will be communicated and in cases where previous experiences or other biases have an impact on the research, it will be discussed thoroughly.

2.6 Research ethics

Research ethics treats the awareness of how to act as a researcher, which is an especially important subsequent are of ethics, since research shape the society long term (Vetenskapsrådet 2017). Myers (2013, p. 45) defines research ethics as *"The application of moral principles in planning, conducting and reporting the results of research studies. A moral stance that involves respect and protection for the people actively consenting to be studied"*. The Swedish Research Council writes that research ethics covers the processes of building up, stimulating and keeping alive the debate about how one should act as a researcher (Vetenskapsrådet 2017), which I believe aligns with Myers' (2013) definition. This topic is divided into the three parts of research ethics that Myers (2013) argues being the most important to acknowledge, honesty, informed consent and permission to publish.

2.6.1 Honesty

This is the key factor within research ethics, because without honesty, all commitments taken within research ethics, such as ensuring anonymity or creation of original knowledge, are worthless (Myers 2013). I intend to be honest describing the research process, and while undertaking, this case study.

2.6.2 Informed consent

The participators of the study should be enabled to give their participatory consent, and also be informed that they have the possibility to, whenever they want, end their commitments to the research (Myers 2013). When conducting a study within an organization, the researcher should seek permission to conduct the study and to perform interviews (Myers 2013). In this case study, I discuss my interview plans with my supervisors to ensure full permission to conduct the study is received.

Myers (2013) acknowledges that sometimes, situations where a trade-off between informed consent and finding out inconvenient truths, can arise. The Swedish Research Council names this type of situations *ethical dilemmas* (Vetenskapsrådet 2017). If there are any ethical dilemmas present carrying out this study, they will be discussed with my supervisors to maintain good relations. Additionally, they will also be represented in the discussion subtitle.

2.6.3 Permission to publish

In order to receive full permission, by the organization, to conduct and publish the study, the interviewees will be anonymized in order to disguise their identities (Myers 2013). Also, appropriate people within the organization will read and approve my work before it is published.

3. Theoretical framework and previous research

This part presents the theory and the previous research that will be used to anchor and analyze the empirically collected material in a research friendly manner. The order of the theoretical subjects is set based on the size of its role in the analysis of the study. The neo-institutional theory plays the main role in this study, ensuring an academically rigid foundation to lean the empirical data against. DevOps is a concept that in this study is used to describe the isomorphic pressure-emerging context of the software developing industry, in sense a complement to the neo-institutional theory. Then the basic meaning of software release management is described, followed by different approaches to technological architectures and their affection on DevOps and software release management. Lastly different approaches of licensing models are described.

3.1 Neo-institutional theory

In the 1950s, theorists started researching the external forces and their affections on organizations (Hu et al. 2007). Earlier literature had focused on the resource-based perspective which assumes that the organization can affect its external legitimacy from within (Suchman 1995). Institutional theory literature however, argues that organizations are social constructions that are influenced by external forces (Hu et al. 2007). The external forces are referred to as institutional effects, which describe how an innovation might first be adopted because of its technical aspects, but in a later stage it is adopted because of its gained legitimacy (Avgerou 2000). After reaching the state where the innovation is considered to be appropriate to adopt because of its legitimacy, pressure will be put on other organizations to incorporate them into their business, to maintain their competitiveness (Hu et al. 2007). The pressure in the mentioned scenario is the institutional effect of the innovation being legitimized, i.e., institutionalized (Avgerou 2000; Hu et al. 2007; Mignerat and Rivard 2009). In other words, when the effects are understood as necessary for operating competitively in an industry market, they are seen as institutionalized (Hu et al. 2007). There can be different types of institutional effects, for example Mignerat and Rivard (2009) argue that institutional effects follow social, technological and legal disruptions. Within information systems research, institutional theory therefore has the potential to help with analyzing how institutional effects influence the application, design and effect of technology, within and between organizations (Mugnerat and Rivard 2009). According to El-Gazzar and Wahid (2013) it has previously been utilized to study information technology innovation, development, implementation and adoption.

When institutional theory was introduced by Selznick in the 1950s, it was mainly definitional and only stated that organizations are affected, not only by internal forces, but also by external ones (Hu et al. 2007). However, more recent work within the subject elaborate on how and why organizations are formed after external pressures (Hu et al. 2007; Oliver 1991; Scott 2008). This newer iteration of institutional theory is named neo-institutional theory, and focuses on the nature and variety of institutional effects, and how they influence organizations (Oliver 1991). Within neo-institutional theory, the main idea is that institutional effects have the effect on organizations, positioned in the same field, to become identical over time, i.e., they become isomorphic (El-Gazzar and Wahid 2013; Hu et al. 2007). Apart from, following industrial disruptions, as mentioned earlier, the institutional effects, i.e., the isomorphic forces, foster the survival and success of organizations during the disruptive era (Hu et al. 2007). To further explore the area, I now present three different types of isomorphic forces; coercive, normative and mimetic, as stated by DiMaggio and Powell (1983) and presented by Scott (2008).

3.1.1 Coercive isomorphism

Coercive isomorphism resembles both formal and informal pressures affecting organizations and their decision makers, most commonly regarding the compliance with governmental regulations and other formal pressures originating from external stakeholders (Mignerat and Rivard 2009). Johnston

(2013) argues that such coercive isomorphic pressures emerge when organizations force less powerful organizations adopt a certain behavior or routine in order to gain legitimacy and comply with rules, policies and regulations. Coercive isomorphism is not only found in governmental or regulative situations, they can be exerted onto an organization by any other organization that possesses the possibility to penalize the submissive organization, should it not be complying (Johnston 2013). Organizations can also be pressured to adopt practices according to cultural expectations from the embracing society, which are seen as informal coercive isomorphism (Hu et al. 2007). Johnston (2013) for example argues that the coercive isomorphic pressure stems from various forms of relationships influenced by power differences, and can differ in formality, direction and origin. In general, coercive isomorphic pressure is exerted on the organization by other organizations or functions with higher bargaining power, for example a parent company, trading partner, governmental function or society (El-Gazzar and Wahid 2013).

In El-Gazzar and Wahid's (2015) study of public cloud adoption in the public cloud sector, the Norwegian data protection authority banned the use of Google Apps as Google's information security requirements did not fulfill the expectations of the authority to be utilized comfortably within the public sector. The ban in El-Gazzar and Wahid's (2015) study is a brilliant example of a formal coercive isomorphic pressure being exerted on an organization. Based on Hu et al.'s (2007) arguing, an example of informal coercive isomorphism could be a society expecting to have online access to medical records, which translates into an exerted pressure on the public health administration to provide it.

3.1.2 Normative isomorphism

Normative isomorphism derives from educational institutions or professional networks and can be present because of professionalism within certain fields of knowledge (Johnston 2013). It is fostered by the professionalization of managers and other influential organizational actors (Hu et al. 2007). Hiring people from other organizations in the industry also encourages normative isomorphism (Hu et al. 2007). Professionalized organizational actors, refers to organizational actors that have similar high-grade education, training and professional network participation, across organizations (Hu et al. 2007). When similarly positioned, professionalized organizational actors often possess are almost interchangeable between organizations because of their similar education and orientation of professional activities (Hu et al. 2007). Within neo-institutional theory, professionalism can be exerted on to an organization by a collective of organizational members that execute actions and behaviors associated with their professional networks (Johnston 2013). Because of their professional similarities, when a new movement in the industrial environment occurs, the professionalized organizational actors across various organizations will see the same potential benefits with adopting or utilizing the core of the movement (El-Gazzar and Wahid 2013). This process lays the foundation for the normative isomorphic pressure of neo-institutional theory, and is often more evident in organizations being in uncertain or ambiguous situations, since its members are then more inclined to look towards their professional environment to find answers and practices the organization can adopt to overcome the ambiguity (Johnston 2013). El Gazzar and Wahid (2015) bring up an example of normative pressure in their study. Because of the lack of experience and knowledge of cloud services, the cloud service adoption rate was low within the Norwegian public sector (El-Gazzar and Wahid 2015). Therefore, a collocation of municipalities was formed with the purpose to exchange knowledge regarding, and foster and promote the usage of, cloud services, in order to ultimately increase the adoption rate of cloud services in the public sector (El-Gazzar and Wahid 2015). The collocation in this case, will act influencing on multiple public organs regarding cloud services adoption, in sense exerting a normative isomorphic pressure onto the organizations.

3.1.3 Mimetic isomorphism

Mimetic isomorphism is the institutional pressure that grows from increasing uncertainty in an industry, and is pressuring organizations to mimic an industry trend or a competitor, to not fall behind in competitiveness (El-Gazzar and Wahid 2013; Hu et al. 2007). It is the feeling of an organization who is inclined to mimic another organization's practices or behavior, because of its success carrying them out (Johnston 2013). Mimetic pressures are more probable to emerge in uncertain environment, mainly because it minimizes the risk when an organization is exposed to an ambiguous situation, but also because mimetic behavior reduces the cost of finding a viable solution (Hu et al. 2007). An example of mimetic isomorphism might be when a small company, without IT expertise, decides to adopt a Software-as-a-service because of its media buzz and a competitor taking the same route, instead of doing research on what the benefits and drawbacks compared to a traditional perpetual license might be. In El-Gazzar and Wahid's (2015) study of cloud adoption, it is argued that the demonstration of a pioneering public organ, successfully showcasing its cloud service adoption, can lead to increased credibility and be a convincing factor for other public organizations (El-Gazzar and Wahid 2015). Thus, the successfulness of a pioneering organization within the industry may exert a mimetic isomorphic pressure towards competitors to adopt similar practices.

3.1.4 Summary

The coercive isomorphism originates from external pressures, formal or informal, such as regulations or culture, the normative isomorphism is based on the similar professionalization of organizational actors across organizations and mimetic isomorphism builds on uncertainty and competitive influences (Scott 2008). Together, these three isomorphic forces constitute the foundation for neo-institutional theory, which can provide insights on how organizational change is driven today (Hu et al. 2007). Because, all organizational change cannot simply be referred to the rationality and decision making of managers (Mignerat and Rivard 2009). Oliver (1991, p. 149) further argues: *"... institutional theorists have tended to focus on conformity rather than resistance, passivity rather than activeness, and preconscious acceptance rather than political manipulation in response to external pressures and expectations"*.

To summarize, isomorphism stresses passive conformity, habits and adherence rather than the intra-organizational managerial power and control, in the context of institutional effects (Oliver 1991). However, this does not mean that neo-institutional theory depreciates the organizational actors' power to determine the direction of an individual organization, it solely recognizes that the decisions carried out by the organizational actors are based on the same beliefs that establish the reaction and structuration of an entire field of organizations (Suchman 1995). Suchman (1995) distinguishes the resource-based perspective from the neo-institutional theory by stating that within the former, the perspective of organizational actors "looking out" is adopted, while in the latter it is the perspective of society "looking in" to the organization.

Hu et al. (2007) argue that neo-institutional theory is an extraordinarily accommodated tool in helping to understand organizational behavior in a fast-paced industrial environment, recognized by continuous change and unpredictability. As previously discussed, this is the type of inter-organizational atmosphere we experience now, and it is fueled by the increasing force of globalization, in association with the exponential growth of constant technological change (Porter and Heppelmann 2014; Eisenhardt 2002; Greenberg et al. 2017). While the isomorphic pressures force organizations to respond accordingly in a strategic manner (El-Gazzar and Wahid 2013), it is also important to acknowledge that in the current fast-paced and uncertain inter-organizational environment, organizations face strategic operational challenges in addition to the institutional effects (Suchman 1995). Suchman (1995) argues that therefore, it is important when analyzing an organization, to look at its strategical response from a perspective of duality, where both the forced

legitimacy effect of isomorphism and the encounter with strategical operational challenges are considered.

In the next chapter, I will therefore present a strategical response framework, conducted by Oliver (1991) and further developed by Mignerat and Rivard (2009) and El-Gazzar and Wahid (2013). Oliver's (1991) take on neo-institutional theory complies with Suchman's (1995) duality argument, by suggesting that organizations do not always acquiesce, i.e., conform to, the isomorphic pressures, sometimes they might also make a compromise, avoid, defy or manipulate as a response to them.

3.1.5 Analytical framework for strategical response to institutional effects

By assuming that the response of an organization, to institutional effects such as isomorphic pressures, can vary in resistance, awareness, proactivity, influence and self-interest, Oliver (1991) produces an analytical framework which can be used to pinpoint the organization's response strategically. She suggests that by recognizing the variety of the organizational response, *"from conforming to resistant, from passive to active, from preconscious to controlling, from impotent to influential, and from habitual to opportunistic"*, the response can be divided into one of five different levels; *acquiescence, compromise, avoidance, defiance, and manipulation* as seen in Figure 4 (Oliver 1991, p. 151).

Strategic Responses to Institutional Processes

Strategies	Tactics	Examples
Acquiesce	Habit	Following invisible, taken-for-granted norms
	Imitate	Mimicking institutional models
	Comply	Obedying rules and accepting norms
Compromise	Balance	Balancing the expectations of multiple constituents
	Pacify	Placating and accommodating institutional elements
	Bargain	Negotiating with institutional stakeholders
Avoid	Conceal	Disguising nonconformity
	Buffer	Loosening institutional attachments
	Escape	Changing goals, activities, or domains
Defy	Dismiss	Ignoring explicit norms and values
	Challenge	Contesting rules and requirements
	Attack	Assaulting the sources of institutional pressure
Manipulate	Co-opt	Importing influential constituents
	Influence	Shaping values and criteria
	Control	Dominating institutional constituents and processes

Figure 4: Strategic Responses to Institutional Processes Framework (Oliver 1991, p. 152).

3.1.5.1 *Acquiescence*

Acquiescence is an organization's conscious, or unconscious, decision to conform with institutional pressures (Mignerat and Rivard 2009). Depending on the consciousness, it can take three different forms or tactics; habit, imitation and compliance (Oliver 1991).

Habit is an unconscious adherence response to taken-for-granted rules or norms (Oliver 1991, Mignerat and Rivard 2009). The habitual response is especially common when institutional norms have reached the status of a social fact (Oliver 1991). Practically, it means that organizations unconsciously respond to institutional norms or practices by reproducing actions that have been carried out historically in the institutional environment (Oliver 1991). For example, an organization can adopt a conventional role distribution, like students and teachers or development and operations (Oliver 1991).

Imitation occurs when organizations, consciously or unconsciously, exercise mimicry of institutional models, similar to the concept of mimetic isomorphism (Oliver 1991; Mignerat and Rivard 2009). It can for example occur as imitation of another, probably successful, organization, or as an acceptance of advice from a hired consultancy (Oliver 1991). For example, adopting a similar innovational process or strategy as a competitor, in the context of uncertainty is referred to as an imitation (Oliver 1991).

Compliance is, in comparison with habit and imitation, a conscious intent to incorporate norms or to conform to institutional requirements (Oliver 1991; Mignerat and Rivard 2009). It mainly differentiates from the other tactics by being a strategical decision to obey a law or accept a norm, taken with the anticipation of self-serving benefits (Oliver 1991; Mignerat and Rivard 2009). For example, when facing a coercive isomorphic pressure created by a trend or an institutional regulation, an organization might conduct a proper study to decide whether or not adoption of the trend or compliance with the new regulation offers enough advantages to be appropriate (El-Gazzar and Wahid 2013). If anticipating that complying with a new regulation will elevate the organization's legitimacy and protect it from the public criticism of resisting, the organization will probably consciously decide to conform to the coercive isomorphic pressure of the regulation (Oliver 1991).

3.1.5.2 *Compromise*

When confronted with an institutional pressure that conflicts with internal organizational goals, an organization might find a compromise to avoid inconsistencies between external and internal demands (Oliver 1991). By carrying out one of the tactics balance, pacify or bargain, the organization promotes its own interests and at the same time conforms to the institutional pressure (Mignerat and Rivard 2009). Compared to acquiescence, compromise leads to a lesser extent of institutional compliance, and the organization's strategic response is more actively produced, stressing their own interests (Oliver 1991).

Balance is an act to attempt to reach conformity between differentiating external stakeholder and internal expectations, especially when conflicting with an institutional pressure (Oliver 1991). By accommodating multiple constituent demands while responding to an institutional pressure, a balancing act is achieved (Oliver 1991). For example, when shareholders demand increased efficiency, the society might express pressure towards allocating resources to corporate social responsibility (CSR), executing a strategic response that finds balance between the two can be fruitful for the organization (Oliver 1991).

Pacifying is quite similar to executing the balance tactic in the sense that it tries to handle multiple stakeholder interests and conform to institutional pressure at the same time (Mignerat and Rivard 2009). Though, to fit within the framing of 'pacify', the response should be part resisting, part

placating (Mignerat and Rivard; Oliver 1991). For example, an organization might show resistance to the coercive pressures of its parental organization, but modify its processes to conform to the pressure, because of the parental organization being the “feeding hand” (Oliver 1991). Another example is a manufacturing organization that, when being pressured to revoke the production of a harmful product, instead chooses to redesign a product to correspond to the expectations (Oliver 1991).

Bargaining is the most active form of conforming to an institutional pressure by compromise (Mignerat and Rivard 2009). By bargaining, i.e. negotiating, with institutional stakeholders, an organization may evoke permit to act outside of the institutional regulation, norm or rule (Oliver 1991). In a sense, bargaining is an active strategic response to promote the organization’s self-interests, instead of passively conforming to an institutional pressure (Oliver 1991).

3.1.5.3 Avoidance

By attempting to circumvent the need to conform to an institutional pressure, an organization tries to avoid it (Mignerat and Rivard 2009). Avoidance can be carried out in three different levels of severity; concealing, buffering or escaping (Oliver 1991).

Concealment is the first of the three avoidance-strategies. It is a response to an institutional pressure that is influenced with falsity (Mignerat and Rivard 2009). By carrying out a concealing tactic as a response, an organization attempts to bypass an institutional pressure by disguising its non-conformity (Mignerat and Rivard 2009). Oliver (1991, p. 154) refers to it as “... an attempt to preclude the necessity of conformity, by disguising the non-conformity behind a façade of acquiescence”. I.e., Oliver (1991, p. 154) uses the expression “window dressing” to describe the activity of an organization establishing plans and procedures to be able to conform to an institutional pressure, though they do not intend to implement them. Within information systems research, an organization might execute a concealing strategic response to the cloud computing market trend by only partially implementing cloud techniques, as a side track with no real focus (El-Gazzar and Wahid 2013).

Buffering is a response by which an organization might circumvent the need to conform to an institutional pressure (Mignerat and Rivard 2009). Buffering is a relatively active avoidance tactic which attempts to reduce the extent of external inspection and evaluation by decoupling internal activities from the external environment (Oliver 1991). It often occurs when public approval of an organization’s production methods is not required for carrying them out efficiently, or when the public approves an organization without the full disclosure of its activities (Oliver 1991). For example, in the clothing industry, the production setting is often decoupled from the end-consumers, leading to a public approval of an organization, despite the Corporate Social Responsibility (CSR)-influenced institutional environment.

Escape is the most active avoidance tactic. It means fleeing from institutional requirements or pressures (Mignerat and Rivard 2009). By changing goals, activities or domain an organization can essentially *escape* to entirely avoid conforming to institutional pressures (Oliver 1991). For example, if a pollution regulation pressures a company to alter its producing mechanisms, it might move the production to another country where the pollution regulations are less restrictive (Oliver 1991).

3.1.5.4 Defiance

Defiance is an active strategic response of resistance to institutional pressures (Oliver 1991). It consists of three levels of rejection to institutional norms, values or regulations; dismissal, challenge and attack (Mignerat and Rivard 2009). In contrast to acquiescence, compromise and avoidance, defiance is more probable to occur when an organization perceives low risk in demonstrating their resistance towards the external element that causes the institutional pressure (Oliver 1991).

Dismissal is a defiance tactic that implies ignoring institutional pressures (Mignerat and Rivard 2009). Organizations are likely to adopt the dismissal tactic when the potential for external justification because of the rejection of institutional pressure, is perceived to be low (Oliver 1991). Sometimes, an organization's understanding of the rationale behind the institutional norm or regulation is not adequate, which can lead to a mindset where the organization tends to reject the institutional pressure by dismissal (Oliver 1991). For example, an organization that does not believe in the validity of the promises with cloud computing might dismiss the cloud computing market trend entirely (El-Gazzar and Wahid 2013).

Challenge. By challenging an institutional pressure, an organization adopts a more active tactic of defiance than dismissal (Mignerat and Rivard 2009). Organizations that challenge institutional pressure sometimes compose a worthwhile uprising (Oliver 1991). For example, there are alternative schools that attempt to counter the traditional, institutional beliefs on how to exercise education effectively (Oliver 1991). In general, organizations will be more likely to adopt the challenge defiance tactic, when they believe to have a more rational and appropriate vision than the institutional environment (Oliver 1991).

Attack. An attacking defiance tactic is distinguished from dismissal and challenge by the active aggressiveness towards the institutional pressure and its originator (Oliver 1991; Mignerat and Rivard 2009). Organizations carrying out this tactic of defiance, aim to attack and assault institutional pressures and the originators of them (Oliver 1991). Attacking is likely to occur when an institutional pressure is directed against a specific organization rather than being general and industry-wide, negative rather than positive or when the organization's privileges and rights are threatened (Oliver 1991). For example, a media channel delivering public criticism might be subject to attack or assault by the organization in various ways (Oliver 1991).

3.1.5.5 Manipulation

The manipulation strategic response is characterized by its activeness and opportunity-seeking intent to alter or exert control over the institutional pressure (Mignerat and Rivard 2009; Oliver 1991). Three different tactics of manipulation can be carried out; co-opt, influence and control (Oliver 1991). The main differentiator of these tactics compared to the other strategic responses is the appreciation of institutional pressures and expectation not constraining an organization's acting, instead the pressure and its source is perceived to be neutralizable, controllable and re-definable (Oliver 1991).

Co-opt. By co-opting, an organization might persuade an institutionally influential constituent to join or support the organization in order to obtain elevated external perceptions (Mignerat and Rivard 2009; Oliver 1991). By demonstrating its dignity and propriety utilizing the co-opt tactic, the organization intends to enhance its legitimacy and thereby attain external approval and neutralize the institutional pressure (Oliver 1991). For example, an organization might co-opt an institutional pressure by attracting an influential constituent to join its board of directors (Oliver 1991).

Influence. Oliver (1991) means that the influence tactic of manipulation is more directed towards institutional belief systems that define acceptance criteria regarding practices or performance. Forming a trade association to lobby against the institutional regulations or to influence public perceptions is an example of the influence manipulation tactic. Another practical example is a coalition of non-profit art organizations lobbying to influence the funding and support from public sources (Oliver 1991). The reason for the influence tactic being possible at all is because often, institutional pressures regarding for example acceptable performance, are in themselves defined by

institutional norms, i.e., they are influenceable by the function exposed to the institutional pressure (Oliver 1991).

Control is the most active tactic of manipulation among the three (Oliver 1991). It is a tactic that tries to dominate the source of the institutional pressure, rather than neutralize, influence or reshape (Mignerat and Rivard 2009). Control is more probable to be carried out when the institutional pressure is perceived as localized or weakly promoted, and especially when the source of the institutional pressure is small in proportion and weight (Oliver 1991). For example, a group of firms might establish their own cloud hosting to achieve full control over their privacy and service quality instead of purchasing an external cloud hosting service (El-Gazzar and Wahid 2013). Oliver (1991) exemplifies the control tactic with a coalition of large corporations that tries to take control, rather than influence, over the way firms approach to CSR is assessed and declared (Oliver 1991).

3.1.5.6 Utilizing the framework

In their study, El-Gazzar and Wahid (2015) utilized Oliver's (1991) analytical strategic response framework in order to identify how organizations within the Norwegian public sector strategically balance their actions as a response to requirements originating from both the governmental and the societal environments. This study is similarly framed by the various isomorphic pressures to identify the presence of different pressures, originating from various internal and external stakeholders, and the extent to which they affect the adoption of DevOps related practices within the studied organization. The next chapter covers the practices, routines and processes related to DevOps.

3.2 DevOps

This part presents DevOps by sweeping through among the current definitions, identifying the aims and important characteristics of the subject, describing how DevOps helps managing organizational disconnect and silo mentality within and organization and lastly presenting what results it might lead to if implemented correctly.

3.2.1 Definition

The definitions of DevOps are widely diverse, describing everything between a set of practices (Claps et al. 2015), an organizational revolution (Smeds et al. 2015) and a business strategy (Farroha and Farroha 2014). While the definitions are diverse, most of the literature seem to refer to DevOps as a concept that emphasizes on collaboration and culture. For example, Colomo-Palacios et al. (2017) defines DevOps as a cultural shift towards collaboration and knowledge-sharing between software development and operations. Kamuto and Langerman (2017) focuses on DevOps as a phenomenon in which the stakeholders of software development collaborate to deliver software continuously. In the same manner, Fitzgerald and Stol (2014) recognizes DevOps as a continuous integration between software and its operational deployment, and Shahin et al. (2017) see DevOps as an approach to solve the traditional disconnect between development and operations by encouraging communication, collaboration and integration between them.

Because of the many different interpretations, DevOps as a term is surrounded with ambiguity (Smeds et al. 2015). That, together with the strong focus on collaboration, I believe is the reason why Farroha and Farroha (2014, p. 288) refer to DevOps as *"a loosely defined set of emerging practices that encourages and enables team work between development and operations"*, one of the definitions I agree with the most. But since I see DevOps as a trending organizational approach within the software development industry, enabling more frequent releases through encouraging collaboration between teams, I will hereafter refer to DevOps as Dyck et al. (2015, Chapter IV) defines it: *"DevOps is an organizational approach that stresses empathy and cross-functional collaboration within and between teams - especially development and operations - in software development organizations, in order to operate resilient systems and accelerate delivery of changes"*.

3.2.2 Aims and characteristics

"If it hurts, do it more often" (Fowler 2018). The main promise of DevOps is to enable more frequent releases (Wettinger et al. 2016; Smeds et al. 2015). Fowler (2018) describes three reasons for why that is desirable. Firstly, most tasks in general become more difficult to accomplish with the growth of its extent (Fowler 2018). Despitefully, it is probable to push the task forward in time, increasing its growth and making it even more painful, as seen in Figure 5 (Fowler 2018). However, if you would carry out the activity more frequently, in incremental steps, the total pain would decrease (Fowler 2018).

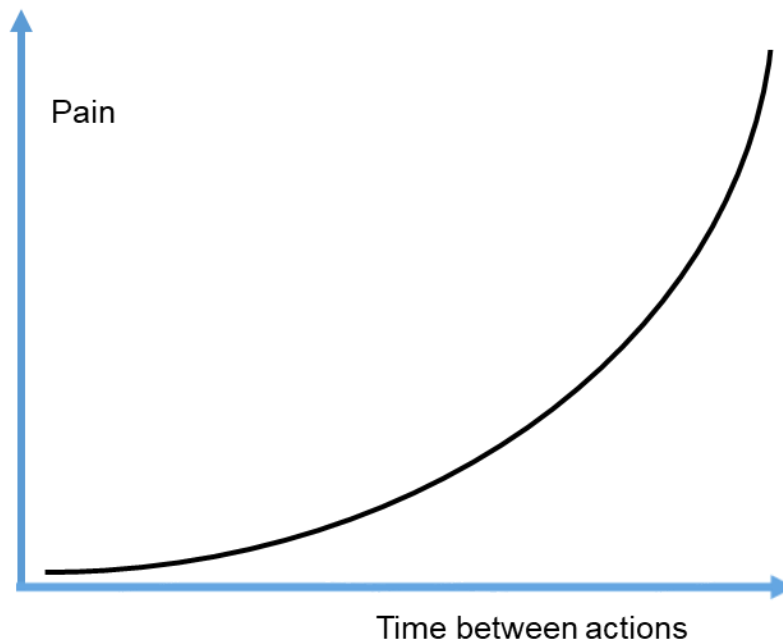


Figure 5: Pain through time between actions (Fowler 2018).

Secondly, increasing the frequency of a task will also increase the possibility for, and amount of feedback (Claps et al. 2015; Fowler 2018). Feedback constitutes a substantial part of the agile software development methodology, because in complex software development projects, it is important to receive continual feedback to identify the current state, and adjust targets (Fowler 2018). Thirdly, the more often you execute an activity, the more effective you get (Fowler 2018). Fowler (2018) relates this to the automation of processes, for example you will be more motivated to automate the process after doing it many times, but at the same time you will understand the process better. By using Fowler's (2018) reasoning, breaking down few large tasks into many small, value can be delivered faster to the customer. This is often seen as the key aim for DevOps (Claps et al. 2015). In the same manner, Fitzgerald and Stol (2014) argue that increasing the frequency of difficult or critical activities, will contribute to overcoming challenging situations.

In order to reach the previously described goal of increasing release frequency, DevOps consists of a few key characteristics. What type of characteristics, and what they stand for varies among literature, but it seems like the characteristics Humble and Molesky (2011) present reflects the opinions of most authors (Humble and Molesky 2011; Claps et al. 2015; Smeds et al. 2015; Dyck et al. 2015). These are:

- Culture
- Automation

- Measurement
- Sharing

3.2.2.1 Culture

The culture characteristic is the most important part of DevOps, it is all about improving collaboration in different ways to streamline delivery, align incentives and reduce the gap between the development and operations divisions (Humble and Molesky 2011). Humble and Molesky (2011) promotes operations employees to take part in the development activities and vice versa. For example, they recommend integrating development representatives whenever production incidents are handled by the operations division (Humble and Molesky 2011). In other words, joint responsibility for delivering software is key (Colomo-Palacios et al. 2017). Wettinger et al. (2016) recommend that DevOps related organizational changes might include the creation of teams including both development and operations personnel. Traditionally, teams are created by the type of activity they work within (Kamuto and Langerman 2017). Kamuto and Langerman (2017) mean that, within DevOps, teams will instead be organized around value streams, to ensure collaboration.

3.2.2.2 Automation

As mentioned previously, short feedback loops are one of the key aims within DevOps (Fowler 2018). In order to achieve that, automation of certain activities, like build, deployment and testing, is required (Smeds et al. 2015; Humble and Molesky 2011). Colomo-Palacios et al. (2017) similarly state that the DevOps approach is heavily based on the automation of software development activities. Humble and Molesky (2011) recommend modeling a deployment pipeline, blueprinting the deployment related processes from the check-in of code to release. Even the provisioning of environments should be automated, according to Humble and Molesky (2011), because then all configuration and recreation steps for correct environments could be maintained centrally, which is simplifying in the case of, for example disaster recovery. Smeds et al. (2015) recognize that, except for streamlining the deployment pipeline, automation also shifts the employees' focus from manual repetitive tasks, to more productive and creative ones, which is beneficial for the organization.

3.2.2.3 Measurement

Measurement is another important characteristic within DevOps (Humble and Molesky 2011). It can be carried out in different ways, but the focus is monitoring key performance indicators (Humble and Molesky 2011). Key performance indicators must be determined with great carefulness, since people act accordingly to how they are measured (Humble and Molesky 2011). For example, Humble and Molesky (2011) recommends measuring the effect of releases on the system stability to achieve more stable systems. By measuring and presenting the key metrics of processes, delivery capabilities will be understood, and proposals for improving the processes might emerge (Colomo-Palacios et al. 2017; Humble and Molesky 2011). As a real-life example, Claps et al. (2015) presents a case, where customer behavior is analyzed in order to align the software development accordingly and thereby shortening the feedback loop.

3.2.2.4 Sharing

Sharing knowledge within the organization, about processes and the tools used, is crucial in order to succeed with DevOps (Colomo-Palacios et al. 2017). Humble and Molesky (2011) identify this sharing on different levels. For example, a simple method of sharing is having the development and operations divisions celebrating releases together (Humble and Molesky 2011). Another method of sharing is communicating the planning ahead, or exercising the planning activity together. For example, to let operations contribute with their activities in the planning of new releases so that new functionality meets the market in the best possible way (Humble and Molesky 2011).

3.2.3 Managing the organizational disconnect and silo mentality

Traditionally, software development is performed with a split between the development and operations divisions and their respective activities (Balalaie et al. 2015). This split is mentioned by several studies within the related literature field and is known by many names; Werner Vogels describes it as the wall between development and operations (Barr 2006), both Dyck et al. (2015) and Wettinger et al. (2016) talk about a barrier between the personnel of the two divisions and others mention the split as a disconnection (Fitzgerald and Stol 2014; Shahin et al. 2017), a rift (Kamuto and Langerman 2017) or the distance between two silos (Colomo-Palacios et al. 2017).

It is generally perceived that this split is problematic for the software release management in many organizations. Humble and Molesky (2011) mean that it is a serious constraint for businesses working towards delivering new functionality faster. Wettinger et al. (2016) says that the split causes long release cycles, and Colomo-Palacios et al. (2017) argue that important problems are rooted in the split, delaying software delivery.

Historically, focus has been put on the software development activity (Fitzgerald and Stol 2014), and when the software product has been finalized, it has been handed over to operations without further thought, or as Werner Vogels at Amazon puts it: *“you take your finalized software to the wall that separates development and operations, throw it over and then forget about it”* (Barr 2006). Even research has had the same kind of narrow focus. The agile software development method for example, was revolutionary when it hit the industry, but it left a lot to wish for regarding activities like deployment, operations and maintenance (Fitzgerald and Stol 2014). Fitzgerald and Stol (2014) see the one-sided research focus as a possible explanation to the split, often found within software development organizations.

Balalaie et al. (2015) identify that the two divisions often have very different views on the frequency of changes, while the development division tends to produce more changes, the operations division wants high stability. Kamuto and Langerman (2017) argue that the split between the two divisions is characterized by a lack of communication, collaboration and coordination, and find that the split is the result of the two divisions' widely misaligned incentives and goals. They develop on Balalaie et al.'s (2015) ideas and say that it is not just differing goals, the two divisions are assessed differently (Kamuto and Langerman 2017). While the development team is assessed by the number of new features, operations' work is evaluated by application stability and server uptime (Kamuto and Langerman 2017). Fish (2018) recognizes that it should be the other way around, if operations was assessed by the number of releases delivered, and development simultaneously assessed by quality and product stability, we would see cooperation between the divisions. Similarly, Humble and Molesky (2011) state that DevOps is about aligning incentives. By integrating goals and aligning incentives, DevOps facilitates a connection between the two traditionally separated silos (Colomo-Palacios et al. 2017). Going further, the DevOps culture together with the emphasized sharing, automation and measurement, addresses the issue to reduce the barriers and solve the disconnection between the development and operations divisions (Wettinger et al. 2016).

There is a consensus regarding that the divide, and silo mentality, between development and operations must be removed (Kamuto and Langerman 2017), but several impediments are identified in the process. In general, each organization's paths towards tapping benefits from DevOps are unique (Smeds et al. 2015). Often, DevOps projects are met with pitfalls like 'lack of interest in the other side' (Smeds et al. 2015), 'resistance to change' because people tend to prefer already established work methods (Kamuto and Langerman 2017) and silo units 'not willing to invest in new approaches' since they believe what they've already got is best practice for reaching their goals (Kamuto and Langerman 2017).

In order to avoid these impediments, top management needs to engage and strategize for DevOps, easing the possible resistances by showing strong commitment and aligning clashing incentives (Kamuto and Langerman 2017). By ensuring that affected teams and employees are motivated based on mutually beneficial rewards, a holistic team mentality for the organization might be seized and sustained (Farroha and Farroha 2014).

3.2.4 Results

DevOps is meant to result in higher release frequency as mentioned earlier, here three different levels of that result are described. Also, final effects and impacts on both the organization and its customers are discussed.

3.2.4.1 *Continuous integration*

Continuous integration is a software developing practice that originates from among others, the methodology of extreme programming (XP) (Stahl and Bosch 2014). Its main function is to enable software developers to integrate their code into the release branch, a centralized repository, continuously (Claps et al. 2015). This enables the software developers to work on the same software product version and to automate the validation testing. Continuous integration is fundamental to enable the success of continuous delivery and continuous deployment.

3.2.4.2 *Continuous delivery*

Continuous delivery is the next step in streamlining the deployment pipeline, and ensures that the validation tested code branch can be built and prepared for deployment in short cycles (Chen 2017). Continuous delivery treats the build phase of the deployment pipeline, and consists of code analyzes, automated integration tests and the build process that prepares the software product for deployment (Chen 2015). Implementing continuous deployment can be problematic and challenging for many organizations (Chen 2017). However, the benefits are in general huge regarding improvements in time-to-market, customer satisfaction, product quality and efficiency (Chen 2017). Continuous delivery is fundamental to enable the success of continuous deployment.

3.2.4.3 *Continuous deployment*

Continuous deployment is the last step in streamlining the deployment pipeline. While continuous delivery prepares the code for deployment (Chen 2015), continuous deployment enables code to be deployed into production more frequently than before (Chen 2017). Continuous deployment as a concept, builds on the deployment pipeline of successful continuous integration and continuous delivery and further establishes an automated end-to-end process, which includes both user acceptance tests and deployment into a production environment. In practice, continuous deployment enables software functionality to be continuously deployed to the customers, which can be perceived as risky (Roudriguez et al. 2017). However, the deployed changes have a smaller impact than with less frequent manual releases (Roudriguez et al. 2017).

3.2.4.4 *Summary of final effects*

According to the three levels of release management process efficiency discussed above, more frequent releases can be achieved by utilizing practices and routines related to DevOps. There are four main final effects, according to Fowler (2018); decreased pain between releases, increased speed and extent of feedback, increased efficiency and innovation and faster value delivery to customers. By increasing the release frequency, overall stress relating to the release management process is relieved, resulting in more satisfied and wellbeing employees (Fowler 2018). Also, the length of the feedback loop is reduced (Fowler 2018), resulting in more and quicker feedback on the delivered products (Claps et al. 2015). Increased release frequency additionally increases efficiency in the sense that when exercising a process repeatedly, typically better results are continuously

attained (Fowler 2018). By being more efficient at carrying out a process, resources can be allocated towards other activities, for example innovating the exercised process (Fitzgerald and Stol 2014). Lastly, value can be delivered more quickly to the customer by releasing more often, something that is important both for the sustaining a positive customer perception, but also from an economic standpoint since the rate of code-inventory capital turnover is heavily increased, enabling investments elsewhere (Fowler 2018; Cockburn and MacGarvie 2009).

3.3 Software release management

Lahtela and Jäntti (2011) define software release management as the people, functions, systems and activities needed to plan, package, build, test and deploy software and hardware releases effectively into production. In the same manner Hoek and Wolf (2003) say that it is the process through which software is made available to, and obtained by, its users. Jansen and Brinkkemper (2006) instead define it as the storage, publication, identification, and packaging of the elements of a product. The three definitions are quite similar. However, Jansen and Brinkkemper's (2006) definition differ in the way that it does not incorporate the users in the software release management process, which I believe to be important because they constitute an important part in the interchange process of an update.

They emphasize a lot on the customer's perceived value and cost being the main reason for implementing software updates and mean that it is interesting that not more software vendors have a structured plan for increasing customer values and reducing customer costs connected to taking an update (Jansen and Brinkkemper 2006). It is surprising since they in their study find that, in average, 70% of a software vendor's revenue comes from existing support contracts, including delivery of releases and new functionality to existing installations, leaving only 30% from new customer installations (Jansen and Brinkkemper 2006). It is apparent that software vendors should invest in making the latest update release as attractive as possible for the customer, both value-wise and cost-wise. These investments will seem large and unsecure at first, but when customers see the added value the payoff will come quickly (Jansen and Brinkkemper 2006).

Hoek and Wolf (2003) believe that very little academic attention has been put on what happens after the software development process, in other words in the software release management subject. They put specific focus on component-based software systems, and preach that the gap between the vendor and its customers could be closed with the help of a software release management support tool, fulfilling the following requirements:

- Inter-componential dependencies should be explicit – even if they cross organizational borders. The dependencies should also constitute the foundation for the support tool's automated implementation processes.
- Releases should be kept consistent – to let processes be standardized.
- The release process should involve minimal efforts for the vendor – processes like release package creation, publication and information should be automated processes.
- A history of retrievals should be kept – to allow the vendor to track which, and how often, the update release packages has been taken.

(Hoek and Wolf 2003)

Other studies within the subject for example find that shorter release cycles lead to companies getting faster feedback about new features and customers benefiting from fast access to new innovations (Khom et al. 2012).

3.4 Microservices and monolithic architectures

The technological architecture of a software system is closely connected with how it is integrated, delivered and deployed within the deployment pipeline (Smeds et al. 2015). Mainly, two different perspectives of the architecture are described; monolithic architecture and microservices (Smeds et al. 2015). A monolithic architecture means that a software product, even if divided into components, possess a lot of internal dependencies, which therefore require a lot of coordination (Balalaie et al. 2015). In the end, a monolithic architecture sequel as a bottleneck to rapid delivery, i.e. build and test, and deployment (Smeds et al. 2015). The microservices architecture instead resembles a more modularized architecture which aims to render software products as a package of multiple small, independently deployable, services (Balalaie et al. 2015).

Even though DevOps can be used for monolithic architectures, a microservices architecture assists in enabling DevOps by promoting the use of small teams (Balalaie et al. 2015), and extends the benefits of the DevOps approach in the organization (Colomo-Palacios 2017).

Because utilizing a monolithic architecture brings delay to the development life cycle, migrating to a microservices architecture bring many benefits (Balalaie et al. 2015). The possible advantages stemming from a migration are increased flexibility, deployment frequency and better overall team structure (Balalaie et al. 2015). In the end, migrating to microservices from monolithic is an extensively resource heavy project, requiring a lot of time, money and work, which will result in shorter delays within build, test and deployment processes (Smeds et al. 2015).

3.5 Software-as-a Service and other software licensing models

Software is available through a few different licensing models, mainly divided into a perpetual model with a maintenance agreement and subscription licensing models, such as Software-as-a-Service (SaaS) and Application Service Provider (ASP) (Choudhary 2007).

- Perpetual licensing model – large upfront initial investment with additional maintenance fees and costs for upgrades (Choudhary 2007).
- Subscription licensing model – No initial investment, instead a monthly recurring fee that covers support, maintenance, updates and hardware management. Basically, the customer is only paying for the possibility to use the software (Zhang and Seidmann 2010). Ultimately, a SaaS subscription licensing model incorporates the software being hosted in the cloud, eliminating all needs for customer engagement (Zhang and Seidmann 2010).
- ASP – an early variant of SaaS where the customer purchased a perpetual software license but the software was run from the vendor's sites in exchange for a monthly fee in exchange for hardware maintenance (Choudhary 2007).

Utilizing the SaaS licensing model puts focus on separating the ownership and maintenance of the system from the usage (Turner et al. 2008). Dubey and Wagle (2007) mean that the SaaS model gives flexibility in choosing a vendor to the customer, and reduces issues and challenges incorporated in the maintenance of the software as that responsibility and the connected costs are transferred to the vendor with SaaS compared to the traditional perpetual license model.

The strongest factor for the SaaS license model breaking grounds on the software market has to do with the offering of up-to-date software at a predictable monthly cost, without the initial investment accompanied with the perpetual license model (Zhang and Seidmann 2010). The SaaS licensing model seem to lead to stronger product development investments in most situations, leading to higher software quality, compared with traditional perpetual licensing (Choudhary 2007). Another

reasons why the SaaS licensing model is getting more increasingly desirable is for example that the data transfer costs are declining, making bandwidth-based services like SaaS more affordable. Also, customers are realizing they have more control over the vendor-customer relationship when they have the possibility to switch to another vendor if the first one does not perform as expected (Dubey and Wagle 2007).

Within the subject of different software license models, Zhang and Seidmann (2010) discuss the optimal way to offer and license software to customers. They study during what conditions a perpetual license, a subscription model or a mixed hybrid variant is best practice (Zhang and Seidmann 2010). Choudhary (2007) is the first to consider and study whether the type of licensing model does affect the vendor's incentives to invest in product development, increasing the quality of the software. There are also studies researching the customer adoption factors for perpetual and SaaS licensing models, something that will be of interest for this study (Xin and Levina 2008).

4. Empirical findings

In this topic, the empirical findings are presented. The structure is set to maximize clearness for the reader and to maintain the line of argument throughout the study. First, the case company, IFS, is introduced. Then the structure of IFS' main product, IFS Applications is presented together with commonly used phrases. Afterwards, a performed pre-study is presented, per interviewee, ending with a short summary. Lastly, the empirical data collected from the main interviews are structured per empirical sub-topic, in order to ensure variation and clearness for the reader, and make for an efficient transit to the analysis chapter.

4.1 Introducing IFS – Industrial and Financial Systems

IFS is an international actor in the software market, mainly with their ERP-product IFS Applications, a large-scale component-based software system that can be configured and customized to accommodate different businesses in various industries. IFS was founded in Linköping in 1983, and has steadily been growing ever since. By 1999 IFS was represented in all continents in the world. In 2015 IFS was acquired by the large Swedish private equity firm EQT, who since the acquisition continually has invested resources into the organization to realize a swift and rapid growth of the company. Today IFS has about 3500 employees, divided into a centralized research and development division (R&D) split between Sweden and Sri Lanka, and consultancy, support and presales divisions in over 60 countries, geographically divided into regions. These regions are much decentralized and possess a lot of decision making power.

The version of IFS Applications released in 2015, named IFS Applications 9, brought a new release concept for updates. Instead of releasing service packs once or twice per year, the new update releases are scheduled to four times per year, or every three months. The update release concept is a part of the strategy *Evergreen*, which focuses on increasing the number of customers running the latest update release versions to ensure the highest perceived quality possible. Related to Evergreen being pronounced a main strategic investment area during 2016, R&D initiated a feedback activity to investigate whether it was possible to increase the frequency of the update deliveries by 50%, and release six updates per year. R&D believed that by increasing the frequency, the size and significance of each update release would decrease, which in turn would make the implementation and validation processes more incremental and efficient. When the feedback returned from the decentralized regions, it showed that the release implementation process is too complex to motivate an increase of the release frequency.

THREE STEPS TO FULL CUSTOMER CHOICE



Figure 6: IFS' 1-2-3 product offering model (IFS sales document).

IFS has its roots in being dynamic and flexible in its product offering. In the 1980's the IFS founders camped in a tent outside their first customer site, to provide the best service possible. Still today, IFS

is very customer focused. For example as seen in Figure 6, their product offering lets the customer decide how IFS Applications will be setup and purchased, an offering that is not very common in the software industry. First, the customer can choose what parts of the IFS product portfolio they want to use. Secondly, the customer can choose how the solution should be hosted, on customer premises or in the cloud. Finally, the choice of payment model is due. Here, the customer can choose between a subscription licensing model, including installation, implementation and support, or the more traditional perpetual license deal. This choice can be done regardless of the previous decision of hosting model. At IFS, the dynamic and flexible customer-oriented model is referred to as the 1-2-3 model.

4.2 IFS Applications structure

In order to clarify for the reader, the structure and complexity of IFS Applications is described in simplified graphical figures, each with an explicating text. After the initial explanation, further clarification is carried out by defining certain IFS related phrases utilized throughout the study.

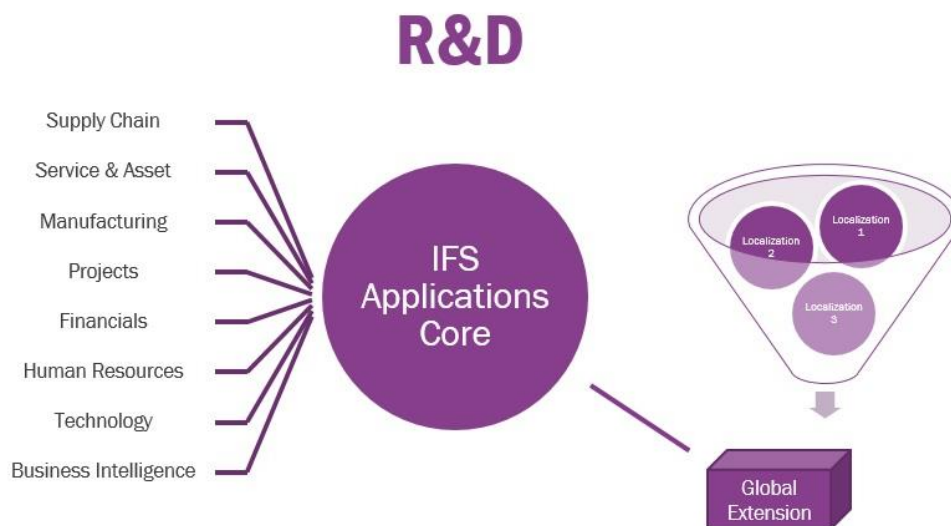


Figure 7: The R&D division's contribution to IFS Applications (Self-produced interpretation).

The IFS R&D division is the foundation of the orientation, product-wise. It consists of eight different sub-divided product organizations, each developing various functionality within its area. For example, the Manufacturing sub-organization develops functionality for IFS Applications that will cover business processes for manufacturing companies, while the Projects sub-organization develops functionality for project-based organization. The Technology division is responsible for developing the technological framework for IFS Applications, and also develops tools and functionality to facilitate, for example, the update release implementation process. The Global Extension division is owned by R&D, but is separated from the other sub-organizations. The Global Extension is relatively newly established and is responsible for the standardization and centralization of regional localizations, which are further explained below. The IFS Applications Core and the Global Extension are the two main products produced by R&D, from a customer perspective.



Figure 8: The IFS regions contribution for IFS Applications (Self-produced interpretation).

The IFS regions mainly consist of consultancy, support and presales sub-organizations. The focus for this study has mainly been IFS Scandinavia, as mentioned previously. In IFS Scandinavia, the sub-organizations are also horizontally divided per industry-orientation. Except for selling, implementing, operating and supporting the customer end-products towards the customer, the regions also develop certain local extensions and localizations that can be used by the customers. A local extension is an add-on product for IFS Applications that for example can add industry specific functionality, to cover specific business flows in, for example, the oil and gas or the retail industry. A localization is extending functionality to cover certain country-specific legal requirements, for example specific governmental tax management requirements. Product-wise, the local extensions and localizations are the regions' main contribution for IFS Applications, but the regions can also implement customizations to the customer's end-product, a very flexible way of modifying functionality to conform with the customer business needs.



Figure 8: An example of how the structure of the IFS Applications customer end-product can look like (self-produced interpretation).

The customer end-product of IFS Applications results in a matrix of the IFS Applications core and the various add-ons. In the example model in Figure 8, the IFS Applications end-product of a global customer active in the construction industry is showed. In this case the matrix consists of the R&D produced IFS Applications 9 core, a construction industry extension with functionality to support specific business flows for construction related companies, the Global Extension to ensure legal requirements for some of the customers active countries are met, an Hungarian localization that yet has not been incorporated into the Global Extension and some customizations based on the customer requirements. Additionally, some integrations and configurations, which can include integrations with some of the customer's other software applications and configurations which for

example can include various permission sets for various user groups within the customer organization.

These various blocks of functionality and add-ons need to be compatible with each other in order to function correctly. With the release of Applications 9, a new layering structure for the code was implemented, called LAA. Explained in a simplified way, the LAA code layering structure ensures that the code of, for example, the local extensions and localizations is not merged together with the code of the IFS Applications Core, facilitating the possibility to swap out the version of a localization to a newer one if required. However, when implementing an IFS Applications core update release, the products in the other layers might have to be reviewed and edited to conform with the IFS Applications core, compatibility-wise. This is my own interpretation of the IFS Applications core structure and functionality, based on multiple interviews, documents and several informal discussions taken place throughout the research study process.

4.2.1 Phrase definitions

R&D

The central developing division of IFS, primarily based in Linköping, Gothenburg and Colombo. R&D consists of multiple sub-organizations, divided product-wise.

Global Extension

An organizational function within the R&D division, responsible for the standardization and centralization of regional localizations.

Regions

Decentralized geographically distributed organizations, commonly consisting of consulting, support and sales sub-organizations. The regions have a very customer oriented approach.

Local Extension

An add-on product for IFS Applications that for example can add industry specific functionality, to cover specific business flows in, for example, the oil and gas or the retail industry.

Localization

An extending functionality to cover certain country-specific legal requirements, for example specific governmental tax management requirements.

IFS Applications update release

A package of security patching and bug fixes for the IFS Applications core released on a fixed quarterly schedule.

IFS Applications major upgrade release

A major upgrade release in the IFS context is when a new version of IFS Applications is released, packed with loads of new functionality and is presented to customers as a separate track. For example, IFS Applications 9 is one major upgrade release, and IFS Applications 10 is another.

Uplifting

The process of ensuring compatibility of a localization, local extension or customization to conform with the implementation of a new update release for IFS Applications.

Evergreen

A strategy that ultimately aims to have all customer end-products always utilizing the latest update release. Evergreen was pronounced as a main investment area during 2016, but was present as a less explicit strategy before that as well.

IFS Managed Cloud

The IFS cloud hosting offering, including the hardware management and distribution of the IFS Applications software. However, depending on the chosen licensing model, support service for IFS Applications can either be included or excluded.

LAA

A code layering structure of the IFS Applications customer end-product. Simplifies the complexity of the customer end-product when implementing update releases.

IFS Update Analyzer

A tool developed by the Technology division within R&D. This tool helps the update release implementation process by automatically analyzing whether or not a new IFS Applications core update release will break compatibility of add-ons like localizations and local extensions, customizations or configurations.

4.3 Explorative pre-study

Three semi-structured interviews were conducted in an early stage to investigate and determine what parts of the software release management are the most challenging and why. The three interviewees all have some sort of role within their regional support organizations, which means they all work in relation to the update concept and, more importantly, they all work closely with the customers. Before the interviews were conducted, a data analysis was made, showing which customers were taking the most and least update releases in each region. By presenting the identified customers to their respective regional interviewee, the thought was that important practices could be identified and disclosed.

In order to ensure proper anonymization, the names of the interviewees are fictive. However, their roles and experience are authentic.

4.3.1 Interviewee Christine

Christine works within the support organization of IFS Scandinavia. She describes the support division as the hub between R&D and the customers, where they have the responsibility to push out the update releases from R&D, but also to report back customer feedback, regarding updates, to R&D. When we discuss the frequency of updates, she explains the current situation.

“There is no customer that can apply updates as often as every three months. They must install, apply and test, and when done, three months has passed and they are supposed to start over again.” (Interviewee Christine, 2018).

She mentions that the testing has been the most painful process for the customers, and states that this probably stands for the largest resistance to taking updates more frequently among the customers. She adds that she understands the resistance.

“Imagine the situation yourself, just having applied a new update and when asking for what to test you receive the answer: ‘There were bugs in a lot of places, you must test everything’” (Interviewee Christine, 2018).

Seldom, the customers have the resources to do that, she says. It takes a lot of time and coordination, for example getting warehouse workers to the IT-division to test and validate their work flows in an updated system environment. The lack of proprietary tools for the customers is also pointed out, and Christine means that if IFS had the responsibility to carry out the acceptance tests, proper tools to facilitate the process would be in place.

Christine identifies how a contradictory relationship between the update frequency and the magnitude of implementation effort emerges. She mentions a customer with some customizations in the application, which complicates the update implementation process.

"The step to take this customer from Update 1 to Update 9 is gigantic. If the updates would have been taken in incremental steps instead, less work would have been required to align the customizations and, more importantly, the risk for something to fail would have been reduced significantly." (Interviewee Christine, 2018).

Christine then describes how important trust in the update concept is for the customers to engage in it. She mentions a customer who received Update 6 for IFS Applications 9 a while ago. The update implementation somehow broke their sales order flow in the live system environment.

"... do you think they will be eager to take the next update?" (Interviewee Christine, 2018).

Apart from the acceptance tests and trust, a big resistance factor for the larger customers to take updates is the amount of coordination needed to be carried out by the support team, to ensure the customer's all extensions and localizations are conformed with, or 'uplifted to', the actual Applications core update release version. Because the ownership of the extensions and localizations often is spread out between multiple regions, it becomes a hassle to ensure the uplifting can be done in time, especially if multiple minor extensions are affected, says Christine.

"Often, when the uplifting coordination project is done for one customer, multiple similar customers are all advised to take the same update to ease the process. At that time the update release might be several versions old." (Interviewee Christine, 2018).

Christine says that they do not communicate with other regions regarding these things, not on her level anyway. She speculates that the reason being the internal thinking of the regions being so different, and having so different types of customer. She ends the interview by envisioning the perfect IFS Applications.

"The IFS Applications in my dreams, enables all customers to always have the latest software available, no customers hanging on to old release versions. That is how it is supposed to be." (Interviewee Christine, 2018).

4.3.2 Interviewee Alex

Alex works within the support organization of IFS Europe West. During the interview he describes the support processes with his three main customers. We start by discussing the release frequency of the update concept, and Alex explains that despite of Update 9 being available, he has one customer that is acceptance testing their Update 7 implementation right now.

"To introduce additional activities at this time, that could cause problems, are out of the question." (Interviewee Alex, 2018).

We talk about the customer trust factor in the update concept, and Alex emphasize how important communication is. For example, he says that nothing in the legal agreements or the contract documents refers to, in any way, how updates are managed. He also mentions Update 1 for Applications 9, a major update causing problems for some customers, which affects the trust factor negatively, especially when there is no communication about it.

“As long as you give customers information ahead of time, they are happy.” (Interviewee Alex, 2018).

When asked about what he would change in the update process if he had endless possibilities, Alex brings up the extensions and localizations. He smiles and says:

“I am quite lucky, neither of the customer projects I run have any localizations.” (Interviewee Alex, 2018).

He explains that because of the lack of an organized, pre-determined way of uplifting local extensions and localizations, ensuring compatibility with recent update releases, it complicates the update implementation process if the customer uses these. He gives a recent example from the past weeks.

“A colleague asked me if I thought it would be possible to lift one of her customers to Update 9. When investigating, it became apparent that the customer had two local extensions that were not going to get uplifted and be compatible with that update. In the end the customer applied Update 8, where the local extensions were available. So, it is really a bottleneck.” (Interviewee Alex, 2018).

4.3.3 Interviewee Emma

Emma works within the support organization of IFS Americas. She has been working within IFS for many years, and therefore has a lot of experience. We initiate the interview by discussing the product offering, especially regarding support contracts. Emma thinks that the offering is relatively similar between different regions, but at the same time she recognizes that there can be differences in the packaging, for example including local extensions in the product offering. She knows that there is some interregional cooperation on her manager's level, regarding standardizing support offerings, ways of work and other routines, but she does not know to what extent.

Emma identifies the acceptance testing as the biggest resistance factor for why customers do not take every update release. She believes it is difficult for the customer to know what to test.

“In fact, they need test everything to ensure quality. Then, one can imagine that it becomes a big hindrance...” (Interviewee Emma, 2018).

Just like the other interviewees, she identifies the problem with local extensions and localizations. She says that a customer installation with some extensions often induce problems when updates are applied.

“Between update releases, not much energy is put into the uplifting of local extensions. Instead, possible problems arising from the incompatibility between the extension and update release are handled if the customer installation crashes during the update procedure.” (Interviewee Emma, 2018).

She thinks that the lifecycle of a local extension should be short, that the extension relatively quickly becomes integrated into the core of IFS Applications, to streamline extension related release management processes.

We then discuss the data analysis I brought to the interview, showing the customers taking the most and the least updates. She asks me if I have confirmed that the best performing customers in the list are in a live, production state, which I have not. She says that the possibility for some of the customers to be in an implementation state, is rather high, and implies that the data then is spoiled with false positives.

“Unfortunately, I do not think it is possible to pick out the status of the installation from the support tool. Sometimes, it is even difficult for me to find out, because we all work differently with the tool.” (Interviewee Emma, 2018).

4.3.4 Pre-study summary

In all three interviews, the wide test scope for the acceptance test validation process seems to be the most distinctive obstacle for receiving and applying update releases more often. All three interviewees also present examples of implementation projects that have been stalled or rejected because of a localization or local extension impacting with the update release being taken. Additionally, the customer trust factor seems to play a very important role in the update concept and is affected by for example inadequate acceptance testing or clashes between an update and local extensions.

Emma implies that there is no apparent focus on the monitoring of support issues. The measurement ability seems to be low when it comes to specific customers and their levels of update releases.

Additional information risen from the pre-study interviews point out that the nature of customers is different across the regions. IFS Americas’ customers most commonly handle their own installations, while IFS Europe West’s customers often utilize partnerships with third party firms. IFS Scandinavia instead seems to offer their consultancy services for implementing update releases at the customer sites. In general, IFS Scandinavia’s customer installations seem to be more customized and utilize more local extensions than installations in the other regions, which are more standardized.

4.4 Main interviews and documents

The results of the pre-study clearly show that there are bottlenecks evident in the release management process, the main ones being the resource-heavy acceptance testing, the lack of localizations and local extensions uplifting and a low customer trust factor. During the main interviews, these bottlenecks were unpacked thoroughly, together with neighboring subjects and DevOps related topics. In excess of the interviews, related internal documents are also reviewed and discussed. The unpacking resulted in 10 areas of special interest, briefly introduced below. The unpacking of the interviews is then structured based on these areas.

Culture and customer focus

This section describes the organization’s workplace culture and long heritage of excessive customer focus. The effect of the new ownership of IFS is also presented here.

Decentralization

Here, the decentralized characteristic of the IFS organization and its advantages and disadvantages are presented.

Localizations and local extensions

In this section, the organization’s approach towards meeting regional-specific requirements is described. The approach has resulted in multiple localizations and local extensions which managerial complicatedness is also presented.

Update release frequency

Here, IFS' approach towards increasing the update release frequency, and the problems regarding it, is presented.

Current market trends and IFS Applications licensing models

In this section, the interviewees' perception of the current market trends of the software industry, and their view of how it affects IFS' customers, are presented. Also, the various licensing models of IFS Applications is stated.

Evergreen and the strategic direction for IFS Applications

In this section, *Evergreen*, one of the main strategic directions for IFS Applications is presented. Also, interviewees' view the organization's work towards realizing Evergreen is discussed.

Technological architecture

Here, the monolithic characteristic of IFS Applications' technological architecture, and the possible effects of breaking down architectural dependencies, are discussed.

Measurement and monitoring

In this section the interviewees' views of the organization's approach to being data-driven by measuring and monitoring tasks and processes is presented.

Automated testing

In this section the interviewees' views of the organization's approach to automated testing and it's possible effects on software quality is presented.

Management's role

Here, the interviewees' perception of the IFS leadership and its role and approach towards steering the organization, determining strategical paths and enabling change is presented.

Before delving down in the empirical results of the interviews, the interviewees are introduced to the reader. In order to ensure proper anonymization, the names of the interviewees are fictive. However, their roles and experience are authentic. Below the interviewees are introduced.

Interviewee Hans

Hans is currently working within R&D. He has a technical background and has been employed for 22 years.

Interviewee David

David is currently working within the presales division of IFS Scandinavia. He has been employed for 18 years.

Interviewee Sofia

Sofia is currently working within R&D. She has been employed for around 20 years and has had multiple different roles within the organization.

Interviewee Jessica

Jessica is currently working within IFS Scandinavia. She has been employed for around 20 years and has had multiple different roles within the organization.

Interviewee Henrik

Henrik is currently working within the technology division of R&D. He has been employed for around 20 years.

Interviewee Victor

Victor is currently working within the technology division of R&D. He has been employed for two years, but has been working with IFS Applications in other aspects before.

Interviewee Anders

Anders is currently working within the technology division of R&D. He has been employed for three years, but has been working with IFS Applications in other aspects before.

Internal documents

Among the reviewed internal documents, the most important are various press releases regarding strategic path decisions and a recent video communication where the newly appointed CEO of IFS discusses his thoughts on IFS and its future.

4.4.1 Culture and customer focus

In its history, IFS has always been very customer oriented. Victor describes a phenomenon I have come across multiple times during my time at IFS. During one of the first customer projects that was carried out, the IFS consultants spent the nights in a tent outside the customer's premises. Since then, IFS has evolved a lot and is now divided into multiple divisions across the world, but according to Anders, IFS still puts a lot of weight in its customers' opinions. The tent is often referred to as a metaphor for the IFS culture of dedication to please the customers.

Victor brings up '@IFS', the name of an old IFS company offering an early form of cloud hosting, often referred to as an ASP (Application Service Provider). @IFS was offered to the customers who wanted it. It enabled the customers to rent the hardware needed to run the software application, and only pay for the actual software, minimizing investment costs.

"While everybody else was selling hardware, IFS focused on its customers by offering @IFS. I think it was back in 1998 or so, probably before cloud was even introduced as a concept!" (Interviewee Victor, 2018).

After some research, a press release describing the @IFS offering was found, dated to May 2000. This is a showcase of the extensive customer focus that is proudly seen as the heritage in the IFS of today. However, the customer focus does not only come with positive aspects. Anders thinks that IFS always lets the customer decide, and that it sometimes leads to difficulties. He believes that the '1-2-3' product offering model has its place, but the various options makes it difficult to support the final customer end-product. Both Victor and Hans have the same opinion.

"IFS is this 'kind Swedish company' that accommodates all people's needs, we need to streamline our offerings to be able to provide qualitative support." (Interviewee Victor, 2018).

"Sometimes, I wish IFS could decide to prioritize and not serve everyone with everything. Right now, we are spreading ourselves too thin." (Interviewee Hans, 2018).

After being acquired by EQT in 2015, Hans recognizes a shift. IFS has acquired companies before, but now, with EQT as owner, the speed and size of the acquisitions has increased dramatically. He says that IFS is no longer a one-product company. Since the acquisition by EQT, Jessica sees a shift in the

mindset of the employees. She thinks that people now are ready for change, they know that money is being invested in the company, they see the growth and await new directions for the organization.

IFS has a great workplace culture, according to Victor and Anders, who are both relatively new at IFS compared to the other interviewees. People are very warm and welcoming, and the culture enables fast paced product development. Hans explains that the humble culture has resulted in the employees adhering for a long time, most of the employees have been within the organization for 15-25 years, and the others are quite recently hired. Hans sees both benefits and disadvantages with the aging workforce. He illuminates the importance of hiring young innovative people with a different mindset and new ideas.

“Code rot with age, especially when it is left untouched. Innovations are continuously needed as the industry environment change.” (Interviewee Hans, 2018).

Product development wise, IFS has accomplished a lot the past years, says Victor and compares with a well-known competitor. However, regarding the exploration of previously untouched areas, the organization leaves a lot to be desired, according to Anders.

“There is a narrow-minded approach present, which becomes more evident the ‘younger’ you are in the organization. Sometimes when I bring up an idea, the response from the older co-workers can be downplaying, like ‘that is not going to work out’” (Interviewee Anders, 2018).

4.4.2 Decentralization

IFS is a typical example of a decentralized organization. The IFS regions are very strong, since they sell the products and rake in the money they are naturally powerful in the decision making, according to Hans. Henrik instead entitles the regions as autonomous, they have the power to act according to their own perceptions. The regions have always had a lot of freedom in their decision making, which really has been the strength of the organization, according to Sofia, who refers to the previously discussed heritage of extreme customer orientation. Both Jessica and David recognize and express that the autonomy induces complexity in the organization.

“The support contracts and sales offering heavily differentiates across the regions, which especially affects our global offering.” (Interviewee Jessica, 2018).

“The regions autonomy delimits us right now, we need to focus more together. With global products with need to have a common global mindset.” (Interviewee David, 2018).

David understands how the competitors easily succeed with their cloud strategies. He means that it is more difficult to set a strategic direction and then head for it, being a decentralized organization. He thinks the key for IFS to succeed globally is to communicate better between regions. The need for better communication is not only evident between regions. Being a software product company, the R&D division plays a significant role in the organization. Henrik argues that for the feedback loop to be correctly closed in the IFS business model, the organization relies on functioning communication between R&D and the regions, the level of which he is not satisfied with today.

We have some forms of knowledge transferring between the divisions, educations for example. But to be fair, sometimes it feels like we rely on somebody else to do the communicating for us. (Interviewee Henrik, 2018)

Hans would like to centralize some parts of the regions responsibilities, for example processes regarding the product offering. He argues that R&D builds a product that is supposed to be delivered

in a streamlined manner, but when it reaches the regions, they sell it in any way they want. Furthermore, localizations, local extensions and customizations are developed and added to the end-product by the regions, which increases the complexity.

Jessica, who works within the region IFS Scandinavia engages the problem from a different perspective. She says that there are too many within R&D working too far away from the customers. She argues that the developers need to understand the end-product and the customer's perspective better.

"... and to achieve that, we need to establish better communication between those who have the knowledge and those who need to take part of it, but that is too difficult because the organizational distance between them is too far." (Interviewee Jessica, 2018).

Hans identifies the organizational divisions as silos with high walls and distance in between. He says that with the silo walls and distance in place, R&D developers will never know how their decisions affect the customers and their end-products. In a similar manner, Sofia brings up an example of the lack in communication and organizational distance. In Hungary, IFS has started to concentrate on selling to smaller customers, which IFS Applications is not developed, nor suited for. In general, R&D has been very product oriented throughout the years, but lately Henrik thinks they have moved closer to the regions and their customers, and that they now see the reality from their perspective as well. He for example argues that the IFS Analyzer, a tool used to analyze the impact of a core update on the installation's existing extensions and customizations, was developed without any inquiry from the regions. Hans argues that R&D see the customer needs from a user perspective, within the subject of user experience. Though, he thinks few have insights regarding the regions' sales, support and consulting activities.

The organizational distance becomes evident when asking the interviewees about differentiating incentives. For example, Anders says that most R&D key performance indicators (KPI's) deal with finding and patching bugs, while Sofia argues how the regions are incentivized to earn income by selling the products and invoicing consultancy hours, not to communicate with R&D.

"For many years we have talked about 'One company', in reality it has been more like 'One logo'..." (Interviewee Henrik, 2018).

Victor and David, while originating from different divisions within the organization both identify an incentive balance problem with IFS' cloud offering, named IFS Managed Cloud. Victor says that there is no real collaboration regarding IFS Managed Cloud, everyone has different responsibilities and nobody has the possibility to steer the direction. David think that the problem with IFS Managed Cloud is that the product owner only has ownership over the product, he does not own any resources.

"Hence, he cannot decide how we in the regions sell, what we sell or when we sell. Instead he has to rely on aligning with all regions and hope they agree with him regarding the strategic direction for IFS Managed Cloud." (Interviewee David, 2018).

Darren Roos, the newly appointed CEO of IFS talks in an internal video communication about the problems induced by the decentralization and autonomy. He identifies how a global customer needs to deal with 8 different IFS regions, who all have their own support offering, consulting contracts, subscription licensing policy and their own way of doing business.

"I want to ensure that we drive a common set of processes and practices about how we do consulting, support, how a partner model looks like, and how we license our software, but still retain and differentiate by staying close to our customers." (Darren Roos video, 2018).

4.4.3 Localizations and local extensions

As mentioned before, both localizations and local extensions are added to the IFS Applications core in the end-product. Localizations constitute of features for country-specific legal requirements such as special tax reporting. Local extensions often consist of industry specific feature packages, for example within construction, retail or oil and gas.

Sofia notices a difference in customer approach towards the localizations and local extensions. She means that customers take functionality covering legal requirements for granted, but when it comes to local extensions, they see it as add-ons that come with an extra cost. Jessica identifies that there is no one that has the power to set requirements for the regions' development of localizations or local extensions. As of now, they do not have to ensure compatibility with the core update levels.

"There is no coordination at all, it is so unprofessional that I cannot believe it." (Interviewee Jessica, 2018).

David recognizes that there have been many internal conflicts regarding local extensions and localizations needing to be uplifted. The internal invoicing model is complicated, which leads to low incentives for carrying out the uplifting process. Jessica means that the regions have no budget for the uplifting process, neither have they the capacity nor the resources required. This is not a new problem, but according to her, it was exacerbated by the implementation of the update release concept (since it is more frequent than the service packs in the past) and it is not looking any better ahead.

"After the release of IFS Applications 10, the regions will have to uplift local extensions and localizations to be compatible with updates for both IFS Applications 9 and 10. It will either double the amount of work, or halve the number of upliftings." (Interviewee Jessica, 2018).

She also recognizes that Scandinavian customers are extra affected by the lack of coordination. Because Scandinavia is a small market, the larger customers are often globally active, thus having the need for multiple localizations. Jessica argues that it does not matter how strategically important or how large the customer is, the regions still have no incentives to perform the uplifting, which affects the customer's possibility to have an updated IFS Applications.

IFS Scandinavia has ownership over a handful of local extensions, mainly within the oil and gas industry. In preparation for IFS Applications 10 a lot of work is put into reducing and streamlining the supply of local extensions. According to Jessica, it has been too easy for sales people and business analysts to call for an extension with new functionality, and then let the consulting team produce it. It has to do with the autonomy of the regions, but also with the lack of collaboration between R&D and the regions regarding demanded functionality, according to Jessica. The regions need to communicate better with R&D to continually merge local extension functionality into core, according to David, who then sees extensions as a possible route for delivering desired functionality rapidly and efficiently.

The issue with localizations has been partly fixed by the establishment of the Global Extension, an initiative that originated from EQT and IFS top management, according to Sofia. The Global Extension is a great way to centralize and solve the uplifting problems with localizations, according to David and Henrik. That Sofia, who is involved in the Global Extension, is positive, says that it is a very welcomed

project for many customers and those responsible for their support. Not all countries' localizations are covered yet, but the plan for Global Extension is to include and cover more countries continuously, according to Jessica. Sofia explains that they have started to encounter problems with the uplifting of the Global Extension. The uplifting process is boring, takes time and requires a lot of resources, according to Sofia. She is worried that the release of Applications 10 will multiply their resource needs for uplifting, since they will have to perform the uplifting activities 8 times per year, instead of 4. At the same time, they have constrained resources budget-wise, resulting in less focus on integrating more localizations into the extension, allocating the resources for uplifting activities instead.

"We are going to get stuck if we are not able to streamline the uplifting activities in some way"
(Interviewee Sofia, 2018).

She is a little bit frustrated because Global Extension cannot utilize the package and release resources within R&D, they have to perform all uplifting related activities themselves. Sofia wants the Global Extension to collaborate closer with the rest of R&D, being treated as one of the other products. She then sees a possibility for the Global Extension to continuously integrate commonly used localizations into core, ultimately focusing on managing fewer, more obscure localizations. Henrik is arguing similarly, meaning that the general uplifting problem for localizations, local extensions and global extension cannot be solved without ultimately integrating and centralizing the localizations and extensions into the core of IFS Applications.

"A global customer's end-product may often be a puzzle of the Global Extension, geographically specific localizations and industry-specific extensions, together forming a very complex matrix. These customers' end-products become very problematic to manage in terms of update releases, especially in the pursuit of accomplishing the Evergreen strategy." (Interviewee Henrik, 2018).

4.4.4 Update release frequency

As discussed earlier, the proposition to increase the frequency of update releases for IFS Applications was declined by the regions. Hans argues that one of the reasons is the lack of understanding in the customer perspective within R&D. He also says it is a mindset thing, the regions do not have to utilize all updates for the frequency increase to create value. For example, if they uplift the extensions and localizations to every fourth update, the update procedure for 'difficult' and 'easy' customers are differentiated in time. Henrik argues that as long IFS has local extensions and localizations that are necessary for the customers, ensuring they are uplifted to be compatible with every update is of great importance.

"Otherwise, R&D could chest-thump about delivering core update releases every week while no customers have the possibility to deploy them." (Interviewee Henrik, 2018).

According to Jessica, the focus needs to be put on reducing the amount of days between the delivery of the core update release and the time it can successfully be implemented in the end-product, with existing local extensions, localizations and customizations. Right now, a global customer first needs to wait until the Global Extension, local extensions and localizations are uplifted, which can take up to two months, and then they can start the acceptance testing.

"When the customers are done with the acceptance tests, another update has already been released from R&D. The customers cannot take this anymore, it is too much of a struggle for them."
(Interviewee Jessica, 2018).

Sofia discusses how the customers often lack enough trust in the update concept in order to take them frequently. They are afraid to take updates often, especially since they have the responsibility to validate it by performing the acceptance testing. The customers feel like they have to perform the expensive and extensive acceptance testing because of inferior product quality, which is understandable according to Jessica. Henrik recognizes that IFS needs to ensure that the customers can test their end-products efficiently, by providing proper tools. In the same manner as Sofia and the pre-study interviewees, Hans also recognizes that customer trust is a very important factor to pursue, when trying to realize the Evergreen strategy.

"I think that the fact alone, that we are starting to worry and think about trust within the subject of Evergreen, is a very good sign. The update concept, in itself, is a step in the right direction." (Interviewee Hans, 2018).

Historically, R&D has only been focusing on the latest product, there were seldom thoughts on things like local extensions or customizations affecting the customer end-product. According to Henrik, since then a big mindset change has taken place. Before the update concept was released together with IFS Applications 9, in 2015, there were service packs that were very difficult to take for customers, says Jessica, who means that the update concept is a great step in the right direction. Similarly, Henrik argues that during the era of service packs there was not the same type of known state for a customer installation compared to today. He says no customer installation was similar to another.

"Every customer installation consisted of a unique combination of single patches for bugs and a service pack or two. Basically, the effort required to apply a service pack was like upgrading to a new version of IFS Applications." (Interviewee Henrik, 2018).

According to Henrik, with Applications 9 and the update release concept, the journey towards Evergreen has started, but there is still a long way to go. The mindset and approach within the organization is still that major functional upgrades are needed outside of the updates. He is therefore fully convinced that there will be another major release of IFS Applications in a few years, called IFS Applications 11. In an update frequency perspective, this type of core upgrade process is problematic since it breaks compatibility for all add-on products and customizations. Jessica argues that, the more core versions IFS supports with updates, the more complex and problematic the situation gets.

If Victor and Anders could choose, R&D would deliver up to 10 updates per day. By identifying and adjusting current bottlenecks, the customers would be able to take as many of the updates they wanted, according to Anders. Victor argues that one major difficulty is all the manual work that is carried out related to each update.

"When there is a new update releasing, it is like the whole organization proceeds into some sort of traumatic condition. I would describe it a form of panic with several deadlines for various groups within different divisions. Everyone perceive this process as very heavy and exhausting, hence why every three months between update releases is seen as frequent enough" (Interviewee Victor, 2018).

Anders means that an increased frequency of update releases is needed for the organization to be able to fix and adjust the evident bottlenecks in the release management process. He thinks rapid and frequent feedback from the customer's update implementation projects, answering the question "Did it go well?" is required to progress further.

"Without connecting and streamlining the feedback loop, I think IFS will slowly start to reverse in Gartner's magic quadrant for ERP enterprises." (Interviewee Anders, 2018).

4.4.5 Current market trends and IFS Applications licensing models

Several of the interviewees identify how the software industry is moving towards a software release model where update releases are non-happenings. Hans for example ironically asks me *“What version of Spotify do you run?”*. He also mentions Microsoft and their transformation to delivering Windows update releases weekly instead of just a few times per year. Henrik means that the mindset Hans is talking about has been around for quite some time. The software industry is moving away from the approach of core versioning. He brings up Windows 10 as an example, which is said to be the last version of Windows. In other words, he recognizes the industry trend, but also adds that the enterprise systems segment, within which IFS is active with IFS Applications, in general is very slowly paced because of high complexity. Though, Hans argues that you cannot be too far behind in an industry and Sofia identifies how various competitors has moved their positions towards the model of continuous update releases, she thinks it is therefore time for IFS to take another step forward, towards realizing the Evergreen strategy.

“Realizing Evergreen is a must, both for us and our customers.” (Interviewee Sofia, 2018).

In the same manner, Hans finds endless possibilities with realizing Evergreen, and the recognition that some of the competitors that are moving in that direction seem to succeed, makes him more eager to see IFS follow.

“It is not just copycatting, the market is maturing and we are not big enough to experiment ourselves, rather we have to pick and choose from the market trends.” (Interviewee Hans, 2018).

Sofia further points out the difference between existing and new customer approaches, and argues that existing customers are more experienced and used to the way IFS manages software releases, while new customers assume they will always get the latest release, without time consuming pitfalls. David, who works within the presale organization of IFS Scandinavia, says that they are planning to offer some sort of package deal of IFS Managed Cloud and a subscription license model to existing customers, in order to promote the possibility of frequent updates, as a step towards Evergreen. Earlier, when they have tried to push for IFS Managed Cloud, many existing customers have perceived it as expensive and not necessary. By packaging it with a subscription licensing, he hopes it can help the customers understand that the additional cost is an investment in a better end-product life cycle.

David continues by explaining how most existing customers in Scandinavia currently utilize a perpetual licensing model, buying the IFS Applications license upfront, consisting of a high initial investment but lower recurring costs. Right now, they are trying to steer these customers to taking more updates by offering an “Evergreen support contract” associated with the upgrade process to IFS Applications 9 or 10. The new support contract adds a recurring cost, in which the delivery, installation and implementation of the update is incorporated. Though, the customer still needs to carry out the acceptance testing themselves.

According to Jessica, new customers are increasingly more often requesting a Software as-a-service model (SaaS). Compared to the IFS Managed Cloud with subscription licensing, the SaaS-model puts the whole operational responsibility on IFS, even the acceptance testing. IFS Scandinavia tries to answer to the demand and has already offered a SaaS-solution to a few customers, according to Jessica.

“SaaS is very hot in the software industry right now. I can understand it, since it is often easier for an IT-manager to get budget acceptance for a recurring cost rather than a larger upfront investment.” (Interviewee Jessica, 2018).

Henrik also recognizes the industry change, he says that earlier, customers have wanted to hold on to older versions for a time to ensure stability. Today, users are more used to rapid releases and swift changes in software.

“Today we are used to Office 365, Spotify and Facebook. ‘Swoosh’, and there is a new button added to the GUI.” (Interviewee Henrik, 2018).

IFS Managed Cloud is an eye-opener for customers, according to Hans. He explains that he talked with a presales consultant who was worried to get security or benchmarking questions by the customer prospects when he started selling IFS Managed Cloud. Instead, he was asked if IFS Managed Cloud ensures always getting the latest update release. He says that with IFS Managed Cloud, customer mentality and demand for Evergreen is fostered. Victor and Anders, who both are involved with IFS Managed Cloud, express how most customers that choose the IFS Managed Cloud service also choose a perpetual license model. They argue that a subscription licensing model, which includes support, updates and upgrades in the price, gives IFS more control over the customer end-products, thus facilitating more frequent updates.

“The fact that we even offer perpetual licensing in the cloud is honestly just stupid.” (Interviewee Anders, 2018).

When David is asked about the licensing cost differences, he acknowledges that perpetual licensing is generally cheaper for the customer when calculating on a product lifetime of at least 4 years. He says that in the industry, a longer life time is usually needed to reach a positive calculation with a perpetual license. According to him, it is a direction set by the management.

“At first, it is easier to sell the subscription license model to new customers, but when they realize that they can pick the perpetual license with for example the Evergreen-subscription for a lower total cost, they often go that route instead.” (Interviewee David, 2018).

Victor speculates that switching the approach to licensing, leaning towards subscription instead of perpetual, would lead to a reduced initial income per new customer. A step he thinks IFS management do not have the courage to take, even though selling a subscription license converts into a much larger income over the customer's lifetime. Jessica did not know about the pricing balance between the subscription and perpetual license models, but she is not surprised when I tell her. She thinks it has to do with perpetual customers being more economically tied to the product, compared to a subscription customer, who has the possibility to cancel the agreement within three months' notice. According to Hans, the focus on perpetual licensing originates from IFS focus on larger customers, where a large upfront investment in general is not a problem. Henrik knows that the subject has been actively discussed and that there is on-going work regarding it. His view is that there are heritage-related reasons behind the imbalance, which takes years to change.

4.4.6 Evergreen and the strategic direction for IFS Applications

Evergreen is one of the main strategic directions for IFS, according to Henrik. His interpretation of the Evergreen concept is that it should be continuously evolving, ultimately enable software that is always utilizing the most recent update release. He thinks it is contradictory to then still have the mindset that there will be a new major upgrade release version at some time. He says that IFS is balancing somewhere in the middle of the path towards Evergreen, having the continually evolving update release concept on one side, and the mindset to promote compatibility-breaking major upgrade versions on the other side. Victor and Anders both want to enable shipping of functionality with the update concept, in that way eliminating the need for major upgrade releases to further

streamline the update release process. Anders argues that, as a byproduct, the customer incentives to apply an update would increase since new functionality in general, is something they want. Hans has a similar opinion, pointing towards Apples iOS operative system, where the users are eager to apply updates in order to acquire new functionality. He mentions a new functionality that will be delivered in 'waves' in IFS Applications 10, together with the first few update releases.

"For example, I believe the Aurena waves we will see in the IFS Applications 10 will increase the adoption rate of the first couple of updates, compared to on IFS Applications 9." (Interviewee Hans, 2018).

David sees that eliminating the major upgrade releases would be a big next step towards realizing Evergreen. He recognizes that the upselling possibility that emerges when a customer is upgrading, would also then be eliminated.

Jessica acknowledges that IFS does not make it easy for their customers to apply updates and follow the Evergreen concept. They like the concept, and often choose IFS Applications thanks to the pledged simple and efficient update release concept, but they find it immature once they utilize it. She means that IFS does not keep their promises regarding the concept. She further argues that the time between the moment an update leaves R&D, to when it can be deployed at the customer's end-product, needs to be shortened. She also mentions the various IFS partners' perspective, when working with a global customer.

"Imagine putting the localizations and local extensions problem in the hands of a partner like Accenture, 'Here you go, this customer is active within 8 different regions and has 17 various localizations and local extensions in 11 countries that needs to be uplifted'. It is an extreme example, but we need to adjust and fix this if we want to be the global actor in the market that we claim to be." (Interviewee Jessica, 2018).

Without being influenced by me, Victor, Anders and Hans all propose a separate customer track, which Anders entitles "IFS Vanilla", to be introduced. This separate track would be fully run in the cloud with a subscription license model, according to Victor. Hans argues that it would require the customer end-products to be fully standardized without customizations. It would be much priced lower than the current offering, and always carry the latest update release, like a true SaaS-model.

"Imagine the customer experience; always receiving the latest update release, very low cost because of the low flexibility and much higher quality thanks to the standardization, dropping the customer's responsibility to carry out acceptance testing. And the short feedback loop it would result in, it would be perfect!" (Interviewee Hans, 2018).

"Then, customers requiring more flexibility and customization would either be able to choose between IFS Managed Cloud for high update release frequency, and global customers requiring a lot of extensions and localizations could choose the current Evergreen track, but to a higher cost which would translate into investments in better coordinated uplifting processes." (Interviewee Victor, 2018).

Jessica who has more of a customer perspective from working within IFS Scandinavia, acknowledges a problem when hearing about the proposition. She argues that a service like that would go against many companies' internet security policies. She would instead like to see an early adopter program for update releases.

"Many of our customers are public companies, they cannot share platform in that way." (Interviewee Jessica, 2018).

David instead wants an IFS Managed Cloud, but without cloud. I.e., an IFS service that runs the customers' end-products on their premises. As a presales consultant, he recognizes the differentiating aspect of the customer orientation and the 1-2-3 product offering. For example, IFS Scandinavia recently won a substantial deal thanks to being able to deliver an on premise setup of IFS Applications combined with a subscription license, which is uncommon in the industry.

In his video communication, CEO Darren Roos identifies that the context of the ERP software industry has changed a lot in the past years and states that the fundamental circumstance needing to be acknowledged is the fact that all customers are moving to cloud based solutions sooner or later, enabling facilitation of more frequent updates. He argues that IFS needs to adhere to that trend.

"Businesses die, because they do not change. When the pace of change outside the organization is greater than the pace of change inside of the organization, the end is near. The industry is changing, our competitors are changing and our customers are changing. If we do not respond to that, we will be redundant." (Darren Roos video, 2018).

4.4.7 Technological architecture

Hans recognizes that sometime in the future, IFS probably has to move their product offering towards cloud and subscription. He says that the technical architecture of IFS Applications is a little bit old and technologically limiting, and therefore it needs to be reviewed in order to be offered as a cloud-first service. He further explains that even though the architecture of IFS Applications is component-based, it is monolithic in the sense that there are a lot of dependencies between the components.

"You cannot just replace a component with a newer version without dependencies failing." (Interviewee Hans, 2018).

He argues that IFS Applications is not based on the concept of microservices, but in the technology division, they try to incorporate characteristics from it. There are many on-going projects relating to the architecture, mainly in an idea stage. Hans argues that in order to reach a fully modular cloud-based IFS Applications, a restructuring into a microservices-based architecture is required. He is very positive regarding the architectural future of IFS Applications, and recognizes that there are many possibilities and a lot to accomplish within the subject.

"I usually joke about IFS Applications being this 'loosely coupled but deeply integrated' system, but I am certain we will reach an architectural level that is based on the concept of microservices in the foreseeable future." (Interviewee Hans, 2018).

Breaking down dependencies and being able to release component by component is an important object in the journey towards Evergreen, according to Hans. He recognizes that it is a major step towards a DevOps friendly architecture, and once again he brings up Microsoft as an example. They not only focused on restructuring their release management processes, rather they have redone the whole architecture for Windows, enabling more frequent and flexible releases. Henrik explains how the LAA code layering concept that was established together with the update concept has made wonders for managing the customer end-products, especially together with the continuously developed IFS Analyzer tool discussed previously.

Hans remembers a project in 2006 when IFS took the first major step towards, what is today known as, DevOps. It was a project that enabled automatic merging of code to a shared repository continuously, i.e., continuous integration. He says it was an initiative that in its whole came from

innovative ideas within the Technology division. Henrik similarly argues that the IFS Update Analyzer project was initiated by ideas and understanding of the complexity of the customer end-product within the Technology division. They both mean that it is those kinds of ideas that need to be in place and facilitated continuously to ultimately realize the Evergreen strategy. Hans finishes by stating that to reach Evergreen, there is more that has to be in place than the right architecture.

“Realizing Evergreen is not just some lines of code and architectural restructuring. Its approaches, principles, processes and an organizational mindset that has to be in place, and that takes time and dedication to establish.” (Interviewee Hans, 2018).

4.4.8 Measurement and monitoring

During the pre-study, it was noticed that the general measurement ability regarding update release levels was somewhat low, with the monitoring reporting a lot of false positives. Jessica confirms that it is not really clear what update the customer is running and whether or not it is a live or test environment that is monitored, when using the support tool. She partly blames the support tool for being old and customized in excess. According to Henrik, the problem is not with the tool itself, rather the lack of established routines for entering data in it. Jessica recognizes the same thing, especially regarding IFS partners and means that it is impossible to get a correct perception of the reality.

“Around 30% of our customers in IFS Scandinavia are run by IFS partners who do not update their information in the support tool at all.” (Interviewee Jessica, 2018).

Victor and Anders have worked together regarding measurement and monitorability in other projects, and they both argue it is of greatest importance when trying to develop and get better in something. They are not surprised hearing about the lack of monitorability in the support tool, because they have experienced a general ignorant mindset to it within the organization. Anders says that all the organization’s KPIs must be based on update levels if the aim is to realize Evergreen. For that to happen, monitorability needs to be ensured and it is the management’s responsibility to steer towards that direction, according to Victor.

Jessica acknowledges that IFS has so many internal systems that are not integrated. She means that monitorability can be in place when customer information is synced between the support tool and other systems, for example the customer relationship management-software (CRM). She argues that there is no value in monitoring data right now, and means that there are more important activities to focus on. Henrik has the same mindset as Jessica, questioning why the monitoring has to be executed at all.

“Why do we even have to measure the levels? Sure, it can be used to see how our product is received, but I think it is more important to focus on developing a good product instead.” (Interviewee Henrik, 2018).

4.4.9 Automated testing

Product quality and the customer trust factor has repeatedly been discussed as important objectives to improve, in the study. According to Henrik, IFS currently utilizes too little automated testing, but large focus is put on expanding it. He also recognizes that automated testing is required in order to ultimately realizing Evergreen. Jessica also sees the need for more automated testing, and argues it is necessary in order to facilitate an increased update release frequency.

Anders identifies that there are many attempts to automate testing flows throughout the organization, but the problem is that these initiatives are scattered like *“little islands”* and are not

collaborating enough with each other. Victor says that he would like to see automated testing flows, ensuring the high quality needed to release functionality incrementally, thus eliminating the need for major upgrade releases. Henrik claims to know why the collaboration is not prioritized within R&D.

“What do you think is prioritized, automated testing and increased quality or new exciting functionality that drives more sales? The dilemma is that quality is not sexy enough to be demo friendly, but everyone expects it.” (Interviewee Henrik, 2018).

Hans visualizes the future of IFS with fully automated testing processes in place and argues that a great amount of possibilities would open up as a result. Loads of resources that would suddenly be available together with the elimination of stress and hard work within the release processes, would lead to a new mindset where innovation would be promoted and facilitated to a higher extent than today, according to him.

4.4.10 Management's role

When talking with Hans about the technological architecture, I ask him if the management are acknowledging the importance. He says that there has been no distinct steering from the management within the subject, thus no additional allocation of resources other than the internal initiatives within the Technology division. Anders expresses that he misses an appointed strategical direction by management, and says that if management for example would communicate *“We are going to offer a IFS Vanilla track”* then people would form and deliver. Victor acknowledges that when you try to lift important business strategy questions, people duck and vaguely point higher up in the hierarchy.

According to Sofia, EQT and the IFS top management were the initiators behind the Global Extension, but now when the Global Extensions important work is being increasingly constrained by the uplifting activities, their resources are limited by management. Jessica identifies the Global Extension's resource constraint as a serious problem. According to her, the localization problem is the most evident bottleneck right now because of the recent launch of IFS Applications 10, which doubles the amount of update releases the localizations need to be uplifted to (IFS Applications 9 + 10). Therefore, she argues, more resources need to be allocated to the Global Extension immediately. She is frustrated.

“I really hope that our new CEO comes in with a new mindset, showcasing that reaching Evergreen is important, because if it continues like this we will be non-existent in the future” (Interviewee Jessica, 2018).

Hans points out that more money is not the solution, rather the investments need to be carried out in a smarter manner, ensuring value creation based on the great functionality of IFS Applications. Victor and Anders both think that there might be people in the hierarchy that are afraid of change, and maybe fear losing their jobs in the process. Anders points out how that kind of behavior needs to be identified and handled with, which can be difficult. Hans speculates that maybe focus will be steered away from IFS Applications by EQT and top management. He means that maybe, the income IFS Applications brings in, will be invested elsewhere, i.e., letting it take the role of a “cash cow”. He explains that he would be hurt by such a direction, and that he is probably not alone with those feelings in the organization.

While Evergreen is pointed out as one of the main strategical targets, the hand that is supposed to determine the pathing towards it, is absent, according to Henrik. In the same manner, Jessica, Sofia, Victor, Anders and Hans also miss a distinctive direction from the top management, and they all have faith in the newly appointed CEO to take action. In his recent internal video communication, CEO

Darren Roos distinctively points out that the IFS organizational structure must be simplified in order to establish globally common practices. According to him, the organization needs to act now to continue to be appealing to its customers in the future.

“This is simply figuring out what we need to do to appeal to our customers. It means we will be more like our competitors, and everyone who operates a global model, but we will differentiate by staying close to our customers.” (Darren Roos video, 2018).

4.5 Empirical summary

In its history, IFS has always been very customer oriented. @IFS is one example from the beginning of the early 2000s, where IFS enabled their customers to rent the hardware needed for running IFS Applications, minimizing their investment costs. This extensive customer focus is proudly seen as the heritage in the IFS of today. Therefore, the flexible “1-2-3” product offering model has its place, but several of the interviewees find that it makes it difficult to support the final customer end-products. Since the acquirement of EQT, IFS is no longer a one-product company and the interviewees notices a shift in the organizational mindset. The employees are prepared for change. The workplace culture within IFS is very warm and welcoming. Some interviewees think the humble culture has resulted in the employees adhering for long times, which also the study’s sample of interviewees indicates. Even though the interviewees identify the importance in hiring young innovative people with a different mindset in a continuously changing industry environment, a narrow-minded approach towards innovations and new ideas coming from newly hired employees, is evident for some of them.

IFS is a decentralized organization with very powerful regions. The regions are often described as autonomous, possessing the power to act according to their own perceptions. The decision-making balance has been the strength of the organization for a long time, but now many of the interviewees recognize the complexity it induces in the organization. For example, support contracts and sales offerings differ across different regions. The interviewees are not satisfied with the amount of communication between the various divisions within the organization. Some of the interviewees identify the organizational divisions as silos with high walls and long distance in between. Often, incentives differ between the siloed divisions. A key for IFS in the future, therefore seems to be improved overall communication and collaboration, both between the regions, and between the regions and the R&D division, which plays a major role in the organization, being an enterprise systems developer. In the region IFS Scandinavia, they feel like the developers within R&D need to better understand the end-product and the customer’s perspective. From R&D’s perspective, it seems like they have moved closer to the regions and their customers, adopting a more realistic approach towards the customer end-products. They newly appointed IFS CEO has identified that the organizational decentralization induces difficulties, especially when working with global customers, and states that he wants to ensure that a common set of processes and practices regarding consulting, support and sales is driven by the regions.

The customers seem to take the covering of legal requirements functionality of localizations for granted while they are prepared to pay extra for utilizing industry-specific local extensions. A problem covering both is the uplifting procedure ultimately would have to be done for every update release in order to ensure compatibility. The interviewees identify that there is no coordination or ownership regarding this process, which therefore naturally leads to internal conflicts. The regions who have the ownership of the localizations and local extensions, lack incentives to perform the uplifting processes, often leaving some localizations or local extensions without compatibility to the latest update release. This problem is exacerbated by the recent major upgrade release IFS Applications 10, since localizations and local extensions now have to be uplifted to be compatible with 8 updates per year instead of 4, when only IFS Applications 9 was supported with updates. This study focuses on the region IFS Scandinavia, and because Scandinavia is a relatively small market it is

often inhabited by globally active customers, utilizing multiple different localizations which makes the uplifting issue extra prominent. The Global Extension is a great way to solve the uplifting problems with localizations, incorporating some of them into a centralized organizational function that has a distinctive ownership. The plan for the Global Extension is to include more localizations and therefore cover more countries continuously, but they have started to encounter the uplifting problem themselves, especially after the release of IFS Applications 10. They are resource constrained budget-wise which means that now, when they have to perform the uplifting activity 8 times per year instead of 4, there is less focus on the scope to include new localizations. Therefore, the Global Extension is trying to be incorporated like one of the other products within the R&D division, to be able to take advantage of, for example, package and release related resources, but they get no positive response.

As I previously discussed, the proposition to increase the release frequency for IFS Applications updates was declined by the regions. Interviewees within R&D point towards the mindset of the regions being the problem, arguing that with more frequent, less massive update releases, and increased coordination would lead to a higher flexibility, both for the regions and their customers. Also, the scope for the customer acceptance testing would be reduced. According to the interviewees, the time between the update releasing and the moment it can successfully be implemented has to be reduced. According to some interviewees, the moment a customer has finished their acceptance testing for an update implementation, another update has already been released by R&D and the implementation process restarts again. It is too much of a struggle for them. Often, customers are afraid to implement updates because they lack trust in the quality and it is their responsibility to perform the acceptance testing. Apparently, customers think that, by performing and funding the acceptance testing, they make up for the inferior end-product quality.

However, the fact that these questions and worries regarding Evergreen are raised is seen as a good sign, according to the interviewees. The update concept is a great step in the right direction, since the previously utilized service packs were very costly and even more difficult to implement for the customers. Some of the interviewees are worried about the mindset and positive approach towards compatibility-breaking major upgrade releases. According to the interviewees, it has to change in order to ultimately realize the Evergreen strategy. Some of the interviewees recognize that the whole organization enters a traumatic state when there is a new update releasing. The release process is perceived as both heavy and exhausting, partly being the reason why the frequency increasing proposition was declined. They interpret that an increased update release frequency could relieve the stress related to the release process, and ultimately fix other evident bottlenecks in the release management process.

Most of the interviewees identify the software industry changing, moving towards a software release model where releases are non-happenings. Microsoft and Spotify are two examples of companies that successfully offers frequent, hassle-free update releases. Both these examples are repeatedly being mentioned by the interviewees. Even though the industry is changing, it is recognized that the enterprise systems segment is usually slower paced because of the high complexity products, but competitors are moving its positions and IFS should not be far behind, according to the interviewees. The customers' approach towards enterprise systems and release frequency are also changing. The interviewees point out that pre-existing customers are used to how IFS manages update releases. However, new customers and customer prospects often assume they will always get the latest update release. In IFS Scandinavia, they are planning to offer a package deal, combining IFS Managed Cloud and a subscription license, to promote the possibility of delivering frequent updates to the customers, as a step towards Evergreen. Currently, most existing IFS Scandinavia customers utilize a perpetual license model, which means an upfront investment that does not include update delivery, installation or implementation. In order to steer these customers to taking more updates, IFS Scandinavia is offering an "Evergreen support contract" that adds a recurring cost, but in exchange it

includes delivery, installation and implementation. Apparently, new customers often request a subscription licensing model, but after when realizing that a perpetual license in general is cheaper when calculating for a product lifetime of more than 4 years, they choose that instead. From an Evergreen perspective, it is negative since a subscription licensing model gives IFS more control over the customer end-product and the update release implementations, enabling more frequent updates.

The IFS journey towards Evergreen has started, and the organization has come a long way on the path there. According to the interviewees, IFS is balancing somewhere in the middle of the path, balancing the continuously evolving update concept on one side, and the mindset and approach to promote compatibility-breaking major upgrade versions on the other. The interviewees argue that as a next step, functionality needs to be delivered with the update releases instead, to eliminate the need for upgrade core releases. Especially new customers like the Evergreen update release concept, and they often choose IFS Applications thanks to it, but IFS does not keep their promises regarding the concept. Many of the interviewees, individually propose a separate customer track, entitled “IFS Vanilla” by one of the interviewees. IFS Vanilla would be fully run in the cloud, with a subscription license model, only covering fully standardized installations to enable always carrying the latest update release. It would offer a streamlined customer experience with low flexibility, low cost, high quality and no need for customer acceptance testing. The newly appointed CEO of IFS, recognizes that the enterprise systems industry is changing. He says that all existing and potential customers, sooner or later, are moving to the cloud, enabling facilitation of more frequent update releases. He argues that IFS needs to acknowledge that trend.

Some of the interviewees argue that the technological architecture of IFS Applications needs to be reviewed in order to ultimately realizing the Evergreen strategy. IFS Applications is seen as a monolithic application, which architectural foundation is component-based, with a lot of influencing dependencies. Breaking down those dependencies is an important object in enabling a more DevOps-friendly architecture, thus facilitating more frequent updates. A major step towards Evergreen was taken by implementing the LAA application layering structure together with the IFS Update Analyzer tool, two initiatives that came from innovative ideas within the Technology division, indicating some understanding of the customer end-product complexity, is present within R&D.

In preparation for the pre-study, a data analysis using extracted data from the support tool as conducted. The results showed specific update levels of companies, in order to enable further investigation regarding possible promoters and drawbacks within the update release concept. Apparently, the data was flawed by false positives, originating from a lack of established routines regarding the entering of information, in the support tool. In general, the focus on measurement and monitorability seems to be low within the organization. According to some of the interviewees, extensive focus on measurement and monitorability is crucial when trying to evolve and develop anything. Others do not think there is enough value in measuring and monitoring the update levels to increase the focus.

The interviewees recognize the need for automated testing activities in order to ultimately realizing the Evergreen strategy. There are initiatives and attempts to utilize automated testing within the organization, but they are scattered and there is no real collaboration present. A possible reason could be that new exciting functionality drives sales better than high qualitative, according to one of the interviewees. Realizing a continuous flow of automated testing throughout the organization would shorten the update release process, eliminate stress and relief hard work within the release process, encouraging innovation instead.

Within many areas in the organization, the experience of the interviewees is that they lack a distinctive strategic direction being communicated by the management. For example, if the IFS

Vanilla idea would be encouraged, people would form and deliver. The Global Extension important work being constrained by resources is another management related issue that induces frustration among the employees. The uplifting problem is identified as a serious bottleneck in the release management process by most of the interviewees, yet they are neither being admitted accompanying the other R&D products to utilize their resources, nor given the budget-allowance to increase their own.

Some of the interviewees believe there could be people higher up in the hierarchy that are afraid of change. Other speculate that the investment focus is moved away from IFS Applications by EQT and the top management, something that could hurt many of the employees who have worked hard with developing the product throughout the years. While Evergreen is pointed out as a main strategical objective, a distinctively determined direction towards it seems to be absent. Many of the interviewees puts faith in the new IFS CEO, who proposes changes and express that the IFS organization must act now to not become insignificant in the future.

5. Analysis

The analysis is primarily based on the identification of significant institutional pressures towards adopting processes and practices from the DevOps concept. The structuring of the analysis chapter is set based on the theoretical framework. The subtopics are entitled after the three main types of isomorphic pressures; coercive, institutional and mimetic. For each of the isomorphic pressures, the distinctness and extent are discussed. Then, the IFS organization's engagements are discussed and evaluated as strategic responses to the isomorphic pressures. Lastly, the analysis is summarized to distinguish the most important segments and points of the chapter to the reader.

5.1 Coercive isomorphism

This section is structured in five subtopics. In “5.1.1 Organizational pre-requisites” the organization's main characteristics; the excessive customer focus and the decentralization, and their effects on the application of neo-institutional theory, specifically coercive isomorphism, is discussed. Then, in “5.1.2 The demand for an efficient update release model”, the demand for a more efficient update release model, originating from a diversity of stakeholders, are first connected to DevOps theory and then the organization's response to the demands is interpreted from a neo-institutional perspective. “5.1.3 The Global Extension” covers an initiative to streamline the update release process by centralizing the majority of the localizations. Its role as a strategic response to the coercive isomorphism is also discussed. “5.1.4 The marketing of Evergreen” discusses the marketing of the update release model for IFS Applications, and its role as a strategic response towards coercive isomorphism. In “5.1.5 Software licensing models” the different software licensing models of IFS Applications, and how the interplay between them can be interpreted as a strategic response to the coercive isomorphism is discussed.

5.1.1 Organizational pre-requisites

The findings point towards IFS proudly executing a customer-oriented approach, originating from a heritage of always being close to the customer. For example, the tent used as a facility during one of the first customer projects, is continuously referred to as a metaphor in both interviews and documents. At the same time, the customer focus seems to sometime lead to difficulties. Both Victor and Hans describe the ‘1-2-3’ product offering model as problematic. Hans believes IFS is “*spreading itself to thin*” by serving every customer with everything. According to neo-institutional theory, formal coercive isomorphism can be interpreted as the isomorphic pressure exerted by internal or external stakeholders on an organization (Mignerat and Rivard 2009). El-Gazzar and Wahid (2013) build upon the formal coercive pressure and mean that it increases with the bargaining power of the stakeholder exerting the pressure. It can be argued that the extensive customer orientation of IFS increases the bargaining power of the customers, since IFS try to accommodate all customers' needs by establishing the flexible ‘1-2-3’ product offering model. Therefore, I argue that any requests or demands expressed by existing or potential IFS customers can be interpreted as coercive isomorphic pressures exerted onto the organization.

The intra-organizational context of IFS constitutes of Evergreen being a pronounced strategy and main investment area, together with the ownership of EQT, who is heavily investing into the organization to enable growth and ultimately a profit in return. During the interviews, it became evident that IFS is a typical example of a decentralized organization. It seems like the regions possess a great portion of the organizational decision-making power, being described as being *very powerful, autonomous, having the power to act according to their own perceptions and enjoying a lot of freedom in their decision making*, by the interviewees. Because of their autonomy and powerfulness, the regions can be argued to possess a significant amount of bargaining power. The Evergreen strategy, the EQT ownership and the regions decision-making power should therefore all have considerable effect on strategic decisions.

5.1.2 The demand for an efficient update release model

In IFS Scandinavia the fact that customers now are demanding an Evergreen update release model is becoming increasingly more evident. Jessica for example notices how customers appreciate, and often choose to purchase, IFS Applications because of how the marketing campaigns appraise the Evergreen update concept as a simple and efficient way to keep the customer end-product up to date, with the latest security patches and bug fixes. Another indicator is that some new customers are requesting a full SaaS offering, something IFS has not been offering before. A SaaS software offering is defined by being hosted in the cloud and paid by a recurring fee that includes the full range of support service, as it is basically a 'rent' of the possibility to use the software, eliminating the need for any customer engagement and ensuring an updated software (Zhang and Seidmann 2010). Both Henrik and Jessica understand the demand, and Henrik refers to customers being used to modern consumer software, e.g., Spotify, which commonly are being delivered as a SaaS. In the same manner, Hans recognizes a changed customer mentality while describing how a presales consultant showcasing IFS Managed Cloud was prepared for questions regarding security and benchmarking, instead the customers asked if IFS Managed Cloud ensured getting the latest update release deployed continuously.

As stated by Johnston (2013) a coercive isomorphic pressure is recognized by the, formal or informal, request or demand by an entity onto a less powerful entity, to comply with and conform to the adoption of a regulation or behavior. Based on the characteristics of a coercive isomorphic pressure (Johnston 2013; Mignerat and Rivard 2009; Hu et al. 2007), and the identification of the IFS organization being submissive to its customers' requests and demands, I argue that the examples of the positive marketing response, the requests for a full SaaS offering and the questioning for continuously delivered updates with IFS Managed Cloud, constitute a coercive isomorphic pressure for the offering of an efficient update release model, enabling the customers to continuously being able to utilize the latest update release, is being exerted onto the organization of IFS.

The organization is struggling with responding to the coercive isomorphic pressure exerted by the customers. The R&D division proposed an increase in update release frequency for IFS Applications in order to conform to the customer demands and the Evergreen strategy, by reducing the size and significance of each update release. According to Fowler (2018), the total pain is reduced if carrying out an activity more often, in incremental steps instead. This is one of the main benefits with adopting a mindset based on the DevOps concept (Fowler 2018). The proposition was declined by the regions because of customers not utilizing the full four per year that was offered before, despite the promised efficiency increase of the update release implementation and validation process. I believe this is an effect originating from the decentralized model of IFS, where, as discussed, the regions possess a significant amount of bargaining-power in the strategic decision making. It also seems like the uplifting problem regarding localizations and local extensions stems from the power balance, according to the interviewees. No-one has enough bargaining power to control and coordinate the uplifting. Jessica notices how the lack of coordination of the uplifting activity is evident regardless of the strategic importance of the customer, and that it inflicts damage on the relationship with customers.

By utilizing the theoretical lens of Oliver's (1991) strategic response theory, the proposition by the R&D division to increase the update release frequency can be seen as an attempt to carry out a compromising response through a balance-tactic, because it was an attempt to reach legitimacy by conforming with the coercive isomorphic pressure of the customers, while at the same *balancing* meeting the regions' opinions and striving towards realizing Evergreen. Instead a defiance response through a dismissal tactic was carried out, because of the regions' *denial* to increase the update release frequency (Mignerat and Rivard 2009). Oliver (1991) argues that sometimes the dismissal

tactic is carried out when the understanding of the isomorphic pressure and its adjacently surrounding circumstances is not adequate. This aligns with Hans' perception of the regions misinterpreting what an increased update release frequency could result in, in other words the fact that they do not have to utilize all updates for the frequency increase to create value.

Hans identifies the R&D division and the regions as silos with high walls and long distance in between them, something that is very common in the software industry (Humble and Molesky 2011). Furthermore, incentives often differ between the divisions according to the interviewees, for example Henrik says:

"For many years we have talked about 'One company', in reality it has been more like 'One logo'..."
(Interviewee Henrik, 2018).

Differing incentives is problematic for reaching a more efficient software release management and in sense, causes long release cycles (Colomo-Palacios et al. 2017; Wettinger et al. 2016). Within DevOps, the cultural aspect with focus on improving collaboration, aligning incentives and eliminating the split through communication, is therefore the most important (Humble and Molesky 2011). According to Kamuto and Langerman (2017), there is consensus in the literature about the divide, and silo mentality, between development and operations must be removed. Many of the interviewees would like to increase the amount of communication within the organization which could reduce the organizational distance.

"We need to establish better communication between those who have the knowledge and those who need to take part of it." (Interviewee Jessica, 2018)

Then, the differentiating views and opinions of R&D and the regions could conform, leading to a reduction, and ultimately an elimination of the conflicting intra-organizational pressures arising that affect the strategic decision making and the possibility to respond accordingly to an isomorphic pressure. By suggesting the centralization of some of the regions' responsibilities, I interpret that Hans wants to reduce their organizational decision-making power. Sofia is also arguing in the same manner when describing the problems IFS Hungary create when they decide to focus on smaller customers. I argue that by centralizing some of the regions responsibilities, the possibility for them to exert any conflicting pressures would be eliminated, within the centralized areas. CEO Darren Roos' recent directing towards harmonizing the spread and distribution of regional processes and practices by sharing practices and processes across the regions, I believe will accomplish both increased communication and harmonization, instead of centralization, which I believe has a similar effect on the release management process. I think the harmonization will lead to a reduction of the regions bargaining-power discussed earlier, hence, opening up for facilitation of a compromising balancing strategic response to the customer's coercive isomorphic pressure ahead (Oliver 1991).

5.1.3 The Global Extension

The initiation of the Global Extension by the owner EQT and the IFS top management, was an attempt to handle the problems regarding coordination and execution of the localization and local extension uplifting activities. The initiative was planned to enable a more efficient update release implementation process, and ultimately facilitate the customers' ability to continuously utilize the latest update release. For example, Sofia identifies the importance of the Global Extension continuously integrating new, previously decentralized, localizations. Through the lens of neo-institutional theory, the Global Extension initiative therefore could have been seen as a balancing tactic of a compromise response by the organization towards the identified coercive pressure to enable a more efficient update release frequency (Oliver 1991). However, the budget-related resource constraints the initiative suffer from (also originating from the management) limits their

ability to perform the integration, and therefore also their ability to facilitate a more efficient update release implementation. Hence, I argue that the initiation of the Global Extension is more of a concealment tactic of an avoidance strategic response to the coercive isomorphic pressure exerted by the customers, because of the disguising attribute of the resource constraints (Oliver 1991). In sense, by the Global Extension initiative, IFS performs a window dressing activity, showcasing how it will conform to the coercive isomorphic pressure, but being influenced by falsity and ultimately avoiding conforming (Mignerat and Rivard 2009).

5.1.4 The marketing of Evergreen

The organization's marketing of the Evergreen update release model as a simple and efficient way for the customers to continuously enjoy the latest update release, can be analyzed in the same manner. The customers often choose IFS Applications based on the pledged update release concept, according to Jessica, but ultimately, the customers find it immature once they utilize it. Using the same 'window dressing' reasoning as previously, this is an example of a concealment tactic of an avoidance strategic response (Mignerat and Rivard 2009).

5.1.5 Software licensing models

While the perpetual licensing model obstructs the efficiency of the update release model compared to a subscription licensing model, because it generally does not include operation and service (Choudhary 2007), the subscription licensing model focuses on separating the ownership and maintenance from the usage (Turner et al. 2008). The subscription licensing model reduces maintenance related issues and problems because the majority of the responsibility and cost are transferred to the vendor, compared to the perpetual licensing model (Dubey and Wagle 2007). Still the interviews indicate that the IFS Applications offering is inclined towards a perpetual licensing model being cheaper than the subscription licensing model when calculating for a lifetime of four years or more (which almost always is the case within the enterprise systems industry). David argues that the inclining towards a perpetual licensing model is determined and set by the management, and there seem to be many possible reasons for it. One is that changing towards favoring a subscription licensing model comes with great shortsighted costs, according to the interviewees. In contrary, Jansen and Brinkkemper (2006) argue that organizations should invest in making taking the latest update release as attractive as possible for the customer and say that even though the investments may seem large at first, the payoff is quick. In IFS Scandinavia they have started to offer an Evergreen subscription service as an add-on to the perpetual licensing model. The subscription service adds a recurring cost to the licensing, in turn incorporating delivery, installation and implementation of the update release to the customer installation. Thus, making the perpetual licensed customer end-product to act like a subscription licensing. With the help of Oliver's (1991) strategic response framework I find that this can be interpreted as a balancing tactic of the compromising strategic response, because of the attempt to reach legitimacy by conforming with the coercive isomorphic pressure exerted by the customers, while at the same balancing the management's licensing model approach and striving towards realizing Evergreen (Oliver 1991).

5.2 Normative isomorphism

This section is structured in six subtopics. In *"5.2.1 Organizational pre-requisites"*, the organization's main property of normative isomorphism, the employees, is discussed. In *"5.2.2 'IFS Vanilla'"*, one of the ideas to possibly solve the update release implementation problems, and its possibility of being recognized as professionalization, is discussed. In *"5.2.3 Compatibility-breaking major upgrade releases"*, the interviewees' views of IFS Application's major upgrade release concept is discussed. In *"5.2.4 Automation, automated testing and the technological architecture"*, the interviewees' views of the organization's approach to automation and automated testing, as well as their views on the technological architecture of IFS Applications, are discussed and connected to DevOps theory. In *"5.2.5 Monitorability"*, the interviewees' views of the organization's approach to measuring and

monitorability are discussed and connected to DevOps theory. Lastly, in “5.2.6 Strategic responses” the previous five sections are condensed and the organization’s responses are thoroughly discussed and interpreted as strategic responses from a neo-institutional perspective.

5.2.1 Organizational pre-requisites

The empirical findings show that there are many ideas and opinions continuously emerging within the organization. It seems like especially the technology division within R&D historically has been capable of facilitating the emerging and development of these kinds of ideas in a good way, according to the interviewees Hans and Henrik who both describe multiple initiatives that has been accomplished after arising as ideas within the sub-organization. The IFS employees identify the industry environment and the general trends that take place, in different ways. During my time in the organization, I have noticed how many of the employees are interested in software development techniques and other relevant topics, for example agile, lean and DevOps. My impression is that many of the employees listen to podcasts, read blogs and books, and attend conferences on the subjects, which all are sources of professionalism (Johnston 2013). In other words, they contribute to new ideas and behavior changes in the organization to solve problems, by being influenced by their professional networks.

5.2.2 ‘IFS Vanilla’

The most prominent imaging of professionalization within this study is the multiple, individual ideas regarding a separate standardized IFS product, utilizing a full SaaS model. This idea was entitled ‘IFS Vanilla’ by one of the interviewees and aims at solving the problems with long release cycles, lack of collaboration, low customer trust factor and wide customer acceptance testing scope by offering a separate product track where the level of standardization is high and the development and operations are tightly linked together. Anders and Victor both recognize how the SaaS model gives IFS more control over the customer end-product and its update release implementation level, which aligns with Zhang and Seidmann’s (2010) arguments. They argue that a subscription licensing model leads to the vendor getting more control over the customer software product and its update level while the customer getting more control over the vendor-customer relationship, because of the flexible nature of a subscription license (Zhang and Seidmann’s 2010). According to the interviewees, ‘IFS Vanilla’ would enable flawless update release delivery and deployment, high quality, rapid feedback, enable a move of the acceptance test responsibility to IFS and ensure extensive collaboration between development and operations, all important aspects and results of DevOps (Fitzgerald and Stool 2014; Kamuto and Langerman 2017; Humble and Molesky 2011; Fish 2018; Fowler 2018). The fact that it derives from multiple sources within the organization, unaffected by each other, makes it clear to me that it is a form of professionalization influenced vision and solution for the evident problems within the organization. For example, one of the interviewees says ‘IFS Vanilla’ would be a side track, to relief the regions of the update release implementation hassle.

5.2.3 Compatibility-breaking major upgrade releases

The interviewees’ identification of the mindset to have compatibility-breaking major upgrade releases being contradictory when strategizing for an Evergreen update release model, is another indicator of professionalism being evident within the organization. Henrik, Victor, Anders and David all identify the ambiguity of the organization’s situation and want to enable shipping functionality with the update releases to eliminate the need of the major upgrader releases, breaking the compatibility of the customer’s end-products and their localizations, local extensions and customizations. Henrik, Victor and Anders for example see a problem with the major upgrade release mindset hindering the evolvement of the update release concept, which I connect to the DevOps literature where Fowler (2018) argues that the more frequently you carry out an activity, the better you get at doing it. What Victor, Anders and Henrik mean is that by breaking the update release concept with a new major upgrade release, including a new update release concept, there is not

enough update release iterations to improve with, compared to an infinite chain of update releases, which can be improved and made more efficient, continuously. David, Victor and Anders also see the increased incentives for the customers to implement updates, given by the integration of functionality in updates. They argue that the customers' desire for new functionality would reduce the negative effects the low customer trust factor and resource heavy acceptance testing currently have on the update release implementation incentives. They all recognize that eliminating the mindset supporting major upgrade releases would be a big step forward, towards realizing Evergreen, but also that it would require a lot from the organization, for example an increased focus on automated testing, to maintain high quality despite the increased size and complexity of each update release.

5.2.4 Automation, automated testing and the technological architecture

When it comes to automation and automated testing, almost all interviewees recognize them as very important pieces that need to be utilized to their full capability in order to enable a more efficient update release model, and reducing the bottlenecks in the release management process. For example the customer trust factor, the uplifting problem with localizations and local extensions and the wide acceptance testing scope could all be eased by enabling increased automation within the build, packaging and deployment processes according to DevOps theory (Smeds et al. 2015; Humble and Molesky 2011; Colomo-Palacios et al. 2017). In the same manner, Hans compares the monolithic technological architecture of IFS Applications, to a microservices based architecture, which is a more modularized architecture of small independently deployable packages (Balalaie et al. 2015). He describes how the Technology division of R&D is actively trying to utilize characteristics from a microservices architecture into the architecture of IFS Applications, and that there are many initiatives regarding increasing the modularity, and breaking down dependencies, of the architecture. In accordance with DevOps literature (Colomo-Palacios 2017; Smeds et al. 2015), Hans clearly recognizes the importance of breaking down dependencies when aiming towards realizing shorter delays in the release management cycle.

5.2.5 Monitorability

Regarding monitorability, another key aspect of DevOps, the interviewees' views are not as aligned. Jessica identifies how it is difficult to follow up what update release versions the customers run, and Henrik argues it has to do with the lack of coordination regarding entering information in the support tool, once again a circumstance originating from the autonomy of the regions. I think Henrik's recognition seems reasonable after using the tool myself. However, neither Jessica nor Henrik find it to be of any major importance to be able to follow up on the customer data. They think there are more important things to focus on. Victor and Anders, instead show an extensive understanding of practices regarding the DevOps concept, and say that measuring and a high level of monitorability is of greatest importance when trying to develop and evolve a process. They recognize that the whole organization of IFS enters a traumatic state of discomfort and stress between releases, something that is commonly curable by utilizing practices and routines from the DevOps concept (Humble and Molesky 2011). Humble and Molesky (2011) state that KPIs need to be determined with great precision and carefulness, since people act accordingly to how they are measured. By showcasing key metrics of processes, Colomo-Palacios et al. (2017) argue that the processes' capabilities can be understood, and proposals for improvement might emerge. In the same manner, Victor and Anders argue that if the aim for the organization is to realize Evergreen, the various sub-organizations' KPIs must be based on the customer update levels. In order to achieve that, monitorability of the update levels must be ensured.

5.2.6 Strategic responses

A normative isomorphic pressure is recognized by organizational members lifting ideas and executing actions and behaviors obtained from their professional networks (Johnston 2013). According to El-

Gazzar and Wahid (2013) this type of external professionalism affect organizations from within, when new movements are occurring in the industrial environment of the organizations. The professionalized organizational actors will see similar potential benefits with adopting the substance of the movement (El-Gazzar and Wahid 2013). I argue that the ideas of 'IFS Vanilla', the elimination of major upgrade releases and a demand for increased focus on automation are all evidence of concepts from the DevOps movement being exerted into the organization of IFS, from within. The monitorability is questionable in this sense, since about half of the interviewees argue it is not an important factor for realizing the Evergreen strategy. Most of the interviewees seem to feel and be influenced by the uncertainty and ambiguity of the situation the organization of IFS is experiencing. For example, Jessica believes the organization is going to stagnate if the uplifting activities for localizations and local extensions are not improved in some way, Anders has a similar opinion regarding the organizational split and distance.

"Without connecting and streamlining the feedback loop, I think IFS will slowly start to reverse in Gartner's magic quadrant for ERP enterprises." (Interviewee Anders, 2018). (Gartner's magic quadrant is the result of a report of the relative positions of a market's competitors).

According to Johnston (2013), normative isomorphism is often more evident in organizations influenced by uncertain or ambiguous situations, because the organizational members then are more inclined to find answers and adoptable practices in their professional environment. In my opinion, the emergence of many similar ideas, all originating from the DevOps movement, in combination with the experienced organizational ambiguity, is a showcasing of a normative isomorphic pressure towards adopting DevOps related behavior and practices, in order to keep progressing towards realizing the Evergreen strategy, being evident within the organization of IFS.

The organizational response to this isomorphic pressure is double-edged. According to Anders, the problem is that there is no distinct strategic direction set by the management for how Evergreen should be reached. He means that if the management for example would buy-in and embrace the 'IFS Vanilla' idea, people would form and deliver. Right now, he thinks the buy-in is absent. In a similar manner, another interviewee recognizes that, when a strategic business question is lifted, people duck and vaguely point towards higher up in the hierarchy. Because of the managerial absence, some of the interviewees even question whether there are people in the hierarchy that are afraid of losing their jobs because of change, and others argue that maybe investments are put elsewhere and IFS Applications is a dying cash cow. In DevOps literature, top management's buy-in is crucial in easing possible resistances and enabling strong organizational commitment, while strategizing for a more efficient release management process (Kamuto and Langerman 2017). Farroha and Farroha (2014) similarly argue that by motivating teams and employees by communicating mutually beneficial rewards and buying in to innovative ideas, a holistic team mentality and drive can be ensured. Related to this is the perception of Anders and Victor, who both are relatively 'young' in the organization, with 3 and 2 years respectively. They often feel like their ideas are being downplayed by others with more experience within the organization, and think it has to do with their lesser amount of years within the organization. There is indication of the professionalized ideas within the organization not being facilitated or exploited, neither from the management, nor from employees with possibility to execute them, regardless of the multitude of the idea's origin. I argue this indication points towards a defiance strategic response, carried out by a dismissal tactic, by the organization. It means the isomorphic pressure is basically ignored (Mignerat and Rivard 2009). According to Oliver (1991) the dismissal tactic is often carried out when the potential penalty for not legitimating with the pressure is perceived as low. A low penalty is probably especially perceivable in a situation where there is a difference in experience or bargaining power between the communicator and the receiver, which I think often can be accommodated to the situation where an idea is rejected.

Regarding the automation and the automatic testing, there seems to be consensus among the interviewees regarding an increased focus on it being needed to move closer towards realizing Evergreen. For example, automated testing is required to eliminate the need for major upgrade releases, according to David, Henrik and Anders. While the interviewees see unexploited possibilities with investing in coordination and evolution of the existing automation initiatives within the organization, there is no strategy for moving forward with it. For example, Hans envisions the organizational stress relief and encouraged innovation, an increased focus on automation would lead to, which aligns with DevOps literature (Fowler 2018; Fitzgerald and Stol 2014). However, there is no evident strategy for the focus on it. One of the interviews state that priorities might be elsewhere.

“What do you think is prioritized, automated testing and increased quality or new exciting functionality that drives more sales?” (Interviewee Henrik, 2018).

In this case it seems evident that the strategic response is defiance, because of the organization's clear rejection of the interviewees' professionalized want for more focus on automated testing. However, it can be questioned whether the organization is carrying out a dismissal or challenging tactic of the defiance response. If Henrik's thought originates from something that has been internally communicated from the management or similar, then the challenging tactic can be confirmed because of the active countering of the normative isomorphic pressure (Oliver 1991). If not, then it is just speculations and there can be any reason for ignoring the pressure, which leads to a dismissal tactic being carried out (Oliver 1991).

Even though the technological architecture of IFS Applications is monolithic, the Technology division within R&D is facilitating ideas and initiatives regarding increasing the modularity of it, by incorporating characteristics from microservices. Employees within the Technology division recognize breaking down dependencies is an important step in the progress towards being able to increase the update release frequency, and therefore encourage the initiatives. According to Hans and Henrik, this is common practice within the Technology division, Hans mentions the first project towards continuous integration in 2006 that was based on initiatives within the Technology division, and Henrik argues that the IFS Update Analyzer tool, which eases the update release implementation, came to life thanks to similar initiatives within the Technology division. They both argue such initiatives need to be valued and facilitated correctly. It seems like, within the Technology division, it is habitual to encourage such innovative initiatives. Hence, within Technology, the organization performs an acquiescence strategic response to the normative isomorphic pressures, by carrying out a habit tactic (Oliver 1991).

5.3 Mimetic isomorphism

This section is structured in two subtopics. Firstly, *“5.3.1 Organizational pre-requisites”* discusses the interviewees' views on the movement of the software developing industry. Then, *“5.3.2 Strategic responses”* discusses if a mimetic isomorphism is evident, and how the organization is responding to it, from a neo-institutional perspective.

5.3.1 Organizational pre-requisites

Several of the interviewees identify how the software developing industry is moving towards a software release model where update releases are non-happenings. Henrik, Hans and David for example repeatedly bring up companies like Spotify and Microsoft in the interviews.

“What version of Spotify do you run?” (Interviewee Hans, 2018).

The literature states that the trend to release software update releases more incrementally and frequently, increases the flexibility of the software, which helps overcoming release management

related challenges that emerges in the current swiftly changing industry environment (Fitzgerald and Stol 2014; Porter and Heppelmann 2014). As an additional effect, we now see companies moving over to capitalizing on subscriptions and incremental releases instead of charging for upgrades (Caray and Macaulay 2018). While discussing the possibility for IFS to follow the trend and eliminate the major upgrade core versioning, Henrik and Hans both discuss how Microsoft is moving away from core versioning, with Windows 10 being the last major upgrade release of Windows (Microsoft 2018). While Henrik describes the mindset behind the Windows 10 update release schedule, Hans recognize how Microsoft have redone the whole technological architecture for Windows to enable the elimination of the major upgrade release core versioning, and increase the flexibility and frequency of the update releases. Apparently, David, Henrik and Hans all recognize the general software industry trend, but many of the interviewees also express how the enterprise systems segment often lags behind in major innovational revolutions, because of their complex nature. However, Sofia notices how IFS' competitors have started to move their positions and she now wants IFS to do the same. Hans, argues that you should not be too far behind when an industry is changing, and by recognizing that some of the competitors who are moving in the direction of delivering more frequent updates seems to succeed, he gets eager to see IFS follow by taking a step towards realizing Evergreen.

"It is not just copycatting, the market is maturing and we are not big enough to experiment ourselves, rather we have to pick and choose from the market trends." (Interviewee Hans, 2018).

As previously discussed, many of the interviewees want to see IFS ship functionality in the update releases, but not only to be able to focus and improve on the update release concept. Hans, for example, points towards Apple's iOS mobile operative system where customers often are incentivized to implement a newer update release thanks to new exciting functionality included.

Mimetic isomorphic pressure is often exerted onto an organization, when in an ambiguous, uncertain situation and at the same time seeing a competitor succeed by adopting certain practices or behaviors (Johnston 2013; El-Gazzar 2013; Hu et al. 2007). A discussion which determined the organization of IFS being situated in an ambiguous, uncertain situation has already been performed in chapter 5.2. The ambiguity was evident because of multiple interviewees experiencing an upcoming organizational stagnation and maybe even reverse, if the circumstances do not change soon. Therefore, Hans previously described opinion regarding IFS needing to *"...pick and choose from the market trends"* because of its relatively small size seems to be a reasonable response, since Hu et al. (2007) argue that conforming to a mimetic isomorphic pressure minimizes risk, when in a situation influenced by ambiguity.

5.3.2 Strategic responses

It is clear that the interviewees are recognizing the software industry shifting, especially Microsoft's transformation of their release management process, and major upgrade release mindset shift, seem to have influenced the interviewees significantly. Now they want IFS to follow, especially when some enterprise systems competitors have started to move their positions. Arguably, the presence of a mimetic isomorphic pressure is evident, but not to the same extent as the coercive and normative isomorphic pressures. There are few responses to the mimetic isomorphic pressures to analyze, probably because it is a secondary pressure, meaning the coercive and normative isomorphic pressures are much stronger. However, CEO Darren Roos identifies that the context of the enterprise systems industry is changing, and determines the context of, and the conditions for IFS, by stating that all customers are moving to cloud sooner or later, thus enabling more frequent update release models. He determines that IFS needs to adhere to that trend in order to not become redundant.

“Businesses die, because they do not change. When the pace of change outside the organization is greater than the pace of change inside of the organization, the end is near. The industry is changing, our competitors are changing and our customers are changing. If we do not respond to that, we will be redundant.” (Darren Roos video, 2018).

I argue CEO Darren Roos’ statement is an official recognition of the ambiguity and uncertainty of the situation IFS is situated in, and it is also an active response aiming towards conforming to the mimetic isomorphic pressure. In other words, I argue it is an intent to an acquiescence strategic response, which will probably be carried out as a compliance or imitation tactic, depending on the activeness of the response. El-Gazzar and Wahid (2013) for example mention that if the organization conducts a proper study before determining to conform, the response can be seen as a compliance tactic, while Oliver (1991) describes how an imitating tactic is carried out while, actively or passively imitating a competitor or trend.

5.4 Summary of analysis

Even though the customer orientated approach IFS exercises has led the organization to the point of today, it is now complicating its strategic orientation. I identify that the customer focused orientation of IFS, gives the customer an overweight in bargaining-power against the organization of IFS, eventually enabling the customer exerting a coercive isomorphism onto the organization (Mignerat and Rivard 2009; El-Gazzar and Wahid 2013). Apart from the customer focused orientation, the intra-organizational context of IFS constitutes of the determined strategy Evergreen, which focuses on ultimately enabling an efficient way for customers to keep their end-products up to date. Also, a clearly distinguishable decentralization is evident within the organization, with autonomous geographically distributed regions, which are powerful from a decision-making perspective.

In the region IFS Scandinavia it is getting increasingly more evident that the customers want an efficient update release model. New customers have started to request a full SaaS offering, something IFS has not offered before. In the same manner, the customers are asking more questions regarding continuous delivery of update releases than security related ones, when purchasing IFS Managed Cloud. The IFS organization being submissive towards its customers, and the increased demand and requests for an efficient update release model together constitute the basis for my argumentation that IFS is being exerted by a coercive isomorphic pressure by its customers (Oliver 1991).

The organization is struggling to respond to the pressure. There was an attempt to carry out a balancing tactic of a compromise response, by increasing the update release frequency, which in accordance to DevOps literature would increase the efficiency of the release model (Fowler 2018), while at the same time balancing the power of the decentralized regions with the Evergreen strategic direction. The proposition was declined by the regions, despite the promised efficiency increase of the update release implementation. Therefore, a defiance response carried out with a dismissal tactic can be identified instead. According to Oliver (1991) the dismissal tactic can sometimes be carried out when the understanding of the isomorphic pressure is not adequate, something that aligns with one of the interviewee’s perceptions of the regions misinterpreting what an increased update release frequency could result in.

I interpret the different understandings and incentives between the R&D division and the regions as very problematic. By establishing increased communication and eliminating the gap between the divisions according to DevOps theory (Kamuto and Langerman 2017), I believe the intra-organizational forces could conform and enable a unified acquiescence towards the coercive isomorphic pressure exerted by the customers. The Global Extension could have been analyzed as a balancing tactic of a compromise response, but because of constrained resources resulting in a

missed main objective of the extension, I interpret it as a concealment tactic of an avoidance response (Oliver 1991). In the same manner, the marketing of the Evergreen update release concept pledges it to be a simple and efficient way to keep the software up to date. Unfortunately, the organization fails to deliver what they promise. Hence, the marketing of the Evergreen release concept is interpreted as a concealment tactic of an avoidance response (Oliver 1991).

The licensing model options of IFS Applications are inclining the customers to choose a perpetual model over a subscription licensing model more often than in the rest of the industry. According to the interviewees, this circumstance is set by the management. Hence, the licensing model offering balance constitute an intra-organizational force. IFS Scandinavia has to balance this force, with the strategy direction of Evergreen and the customers' demands for a more efficient update release model. Therefore they offer a subscription service to the perpetual licensing model, which adds a recurring cost to the contract, and includes delivery, installation and implementation of the update releases, ultimately increasing the incentives for customers to implement updates. This is interpreted as a balancing tactic of the compromising strategic response, because of the attempt to reach legitimacy by conforming to the coercive isomorphic pressure exerted by the customers, while at the same balancing the management's licensing model approach and striving towards realizing the Evergreen strategy (Oliver 1991).

The empirical findings also show that there are many ideas and opinions emerging within the company to solve the bottlenecks within the release management. One of the ideas, 'IFS Vanilla', continuously arises during the interviews. It is a standardized side track, influenced by DevOps practices, enabling frequent update releases, high quality and a low price. Another opinion that keeps rising is a possible elimination of major upgrade releases, because of it being contradictory when strategizing for Evergreen. Also, increased focus on automated testing recognized as crucial in order to manage with the current bottlenecks. Evolving the technological architecture also appears as a possible enabler for a more efficient update release model. The fact that various interviewees, unaffected of each other, bring up similar ideas that can help the organization on the journey towards Evergreen, for example the elimination of the upgrade core releases or the increased focus on automation, makes me interpret a distinct normative isomorphic pressure being evident. A normative isomorphic pressure is recognized by organizational members lifting ideas and executing actions and behaviors obtained from their professional networks (Johnston 2013).

The organizational response to this normative isomorphic pressure is double-edged. There is no distinct strategic direction by the management for how the Evergreen strategy should be reached, also there is no evidence of a management buy-in into the DevOps influenced ideas neither regarding the 'IFS Vanilla' customer side track, nor the elimination of major upgrade releases. Some of the interviewees even question whether there are people in the hierarchy that are afraid of change, and others argue that maybe investments are put elsewhere than in IFS Applications. In DevOps literature, top management's buy-in is crucial in facilitating strong organizational commitment, while strategizing for a more efficient release management process (Kamuto and Langerman 2017). There is no indication of any of these kinds of ideas being facilitated by the management or any other employees with the power to do so. Therefore I argue, this lack of indication points towards a defiance strategic response, carried out by a dismissal tactic, which means that the organization is responding by ignoring the isomorphic pressure (Oliver 1991).

Regarding the lack of focus on automation and automated tests, increased collaboration between the scattered initiatives within the organization is needed in order to move closer towards realizing the Evergreen strategy. For example, more focus on automated testing is needed to eliminate the need for major upgrade releases. One interviewee explains that the lack of focus may be an active decision by management because increased quality is not 'sexy enough' and does not drive sales in the same way as new functionality. If it can be confirmed to be an active decision, then the response can be

interpreted as challenging tactic of the defiance strategic response, if not then it is interpreted as a dismissal tactic instead, because of the neglect (Oliver 1991). Based on the findings, within the Technology division, the organization seems to be good at facilitating ideas and initiatives stemming from normative isomorphic pressures. The interviewees mention a few projects that have started as innovative initiatives, and today compose great value for the organization. This continuous encouragement of new ideas and initiatives is interpreted as a acquiescence strategic response carried out by a habit tactic, because of the habitual aspect of the encouragement.

The current industry changes are identified by the interviewees. They often point towards companies like Spotify or Microsoft when discussing the industry. Microsoft is perceived to have performed a large transformation of their mindset and technological architecture to enable an efficient update release model, with eliminated major upgrade releases. However, it is identified that the enterprise systems segment of the software industry is commonly lagging behind a little bit because of the complex nature of the software. Some of the interviewees see how competitors within the segment are moving their positions towards a more efficient update release model, and mean that it is time for IFS to move as well.

The IFS organization can be determined as being situated in an ambiguous, uncertain situation. The ambiguity is considered evident because of multiple interviewees experiencing an upcoming organizational stagnation and maybe even reverse, if the circumstances does not change soon. According to Johnston (2013), El-Gazzar and Wahid (2013) and Hu et al. (2007), mimetic isomorphic pressure is often exerted onto an organization that is in a situation influenced by ambiguity and uncertainty, while at the same time seeing a competitor succeeding with adopting certain practices or routines. The presence of a mimetic pressure can be argued being evident in this situation, but it is definitely not as clear as with the coercive and normative isomorphic pressures. The recent statement by CEO Darren Roos regarding the industry change is a recognition of the ambiguity and uncertainty of the current IFS organization's situation. However, it cannot yet be interpreted as a response.

6. Conclusions and discussion

This topic presents the conclusions drawn from the study, answers the research questions and then discusses the contributions, both from a practical and theoretical perspective. First, the fulfilling of the purpose is discussed, then the answers to the research questions are presented and lastly, the contribution is discussed.

6.1 Brief introduction to the conclusion and discussion

The purpose of this study was to explore how and why the organization of a Swedish enterprise systems vendor, interpret and respond to the isomorphic pressures created by the DevOps paradigm within the software industry environment. By interviewing ten people from the organization about bottlenecks in the release management process and mapping their answers to the DevOps paradigm, neo-institutional theory and the strategic response framework by Oliver (1991) could then be used as an outline for fulfilling the purpose.

The first part of the purpose was to identify the different coercive, normative and mimetic isomorphic pressures, currently being exerted onto the organization. To successfully identify the isomorphic pressures, the empirical data was divided into ten themes and then analyzed with both DevOps literature and neo-institutional theory, to understand and position the various isomorphic pressures correctly. The second part covers interpreting and understanding the organizations response to the various isomorphic pressures, in accordance with Oliver's (1991) strategic response theory. The focus during the interviews was to identify the intra-organizational forces and demands that were evident during the study. These forces affect the way strategic responses can be carried out and were important to identify, in order to understand the organization's strategic responses correctly.

6.2 Identifying the presence and extent of isomorphic pressures

The first research question deals with identifying the presence and extent of isomorphic pressures exerted onto the organization. In Figure 9, the identified pressures and their origins are presented.

Type of isomorphism	Description	Source
Coercive	<ul style="list-style-type: none">- Customers like, and often purchase IFS Applications thanks to, the marketing of the efficient update release model.- New customers are requesting a full SaaS offering, to ensure an updated software.- Customers are interested in continuous update releases, rather than security when purchasing IFS Managed Cloud.	Customers
Normative	<ul style="list-style-type: none">- The multiple arising ideas regarding 'IFS Vanilla'.- Employees recognizing contradictory relationship between strategizing for Evergreen and having a positive mindset about compatibility breaking major upgrade releases.- The apparent recognition of increased focus automation and automated testing being crucial for managing the release management bottlenecks and for the advancement of the organization.- Employees identifying that the technological architecture of IFS Applications is not optimal for realizing Evergreen.	Employees
Mimetic	<ul style="list-style-type: none">- Several interviewees identify the changing software developing industry, for example Spotify's non-spectacular update releases are continuously mentioned.- The interviewees are impressed by Microsoft's redoing of the Windows 10 architecture, to eliminate major upgrade releases.	The software developing industry, Microsoft and competitors

	- Even though the enterprise systems segment usually moves slowly in the context of innovational revolutions, some of the interviewees recognize competitors moving their positions, which makes them eager to see IFS do the same.	
--	---	--

Figure 9: Table of identified isomorphic pressures (Self-produced)

The findings show that IFS' extensive customer focused orientation increases the bargaining-power of the customers, evidently defining any of their common demands or requests as coercive isomorphic pressures. The interviews pointing towards the customers purchasing IFS Applications because of the efficient update release model, requesting a full SaaS offering, to ensure efficient update releases and showing interest in continuous update releases, rather than security when purchasing IFS Managed Cloud therefore indicates that a coercive isomorphic pressure is being exerted onto the organization by its customers.

The many continuously emerging ideas on how to deal with the release management bottlenecks being evident within the organization, points towards a substantial amount of professionalization from external DevOps influencing professional sources and networks. The multiple separate ideas regarding an 'IFS Vanilla' separate track, offering a full SaaS model to ensure rapid feedback and continuous update releases, is a main indicator. Also, the recognized contradictory relationship between strategizing for Evergreen and the promotion of compatibility breaking major upgrade releases, the want for an increased focus on automation and the identification of the IFS Applications technological architecture not being optimal for realizing Evergreen are all indicators of a DevOps inspired professionalization becoming evident within the organization. The emergence of many similar, DevOps related, ideas, while the organization is ambiguously trying to handle bottlenecks hindering the efficiency of the release management process, I argue is evidence of a normative isomorphic pressure being evident.

The interviewees identifying the general change of the software developing industry where update releases are non-happenings is an indicator of a mimetic isomorphic pressure being evident. In the same manner, the recognition and positive impression of Microsoft's transformation of Windows and the glancing towards competitors' actions, shows that there is a mimetic isomorphic pressure exerted onto the organization, but probably not to the same extent as the other pressures.

I find it arguable that the DevOps paradigm consists of both a coercive isomorphism of an increasing demand from customers to utilize an efficient update release model, to always get their hands on the latest functionality or bug fixes, and a normative isomorphism from employees who see possibilities to adopt DevOps related practices and behavior in order to meet the increased demand from customers.

6.3 Interpreting IFS' engagements as strategic responses to the identified isomorphic pressures

Strategic response	Tactic	Isomorphic pressure	Empirical findings	Reason
Acquiescence	Habit	Normative	- The Technology division within R&D facilitating internal ideas and initiatives regarding the technological architecture.	- Efficiency
Compromise	Balance	Coercive	- IFS Scandinavia offering an "Evergreen" subscription service for customers with perpetual licensing models	- Legitimacy

Avoidance	Concealment	Coercive	- The Global Extension was supposed to solve a bottleneck and enable a more efficient update release implementation process, but now instead is struggling with uplifting activities	- Resource constraints
Avoidance	Concealment	Coercive	- The marketing of the IFS Applications pledging an efficient update release implementation, but the organization fails to keep its promise.	- Window dressing activity
Defiance	Dismissal	Coercive	- The regions misinterpreting the advantages of an increased update release frequency and declining the proposition	- The autonomy and bargaining-power of the regions
Defiance	Dismissal	Normative	- The absence of facilitating professionalized ideas and solutions to evident bottlenecks within the release management.	- Lack of management's buy-in

Figure 10: Table of identified strategic responses to isomorphic pressures.

As seen in Figure 10, the coercive pressure was the most evident among the different responses, followed by the normative isomorphic pressure. However, most responses are interpreted as being avoiding or defying in its strategic characteristic, and both dismissal and concealment are common tactics, indicating that many initiatives and possible solutions are either ignored or rejected by the organization. From a more holistic perspective, the relatively new update release concept that came with IFS Applications 9 is a big step in the right direction. A mindset change has taken place since the era of service packs where the customer installations had no clear states and were not similar to another. Today, IFS is balancing in the middle of the path towards Evergreen, wanting to evolve the update release concept but also having the mindset of promoting compatibility breaking major upgrade releases. I interpret this as a compromising response, carried out by a balancing tactic according to the strategic response framework.

The interviewees' ideas and solutions to bottlenecks are in general not facilitated within the organization. The employees seem to feel an absence of a management buy-in, some even think that people in the hierarchy are afraid of change because of this. Perhaps, there is also a lack of model for communicating and sharing these ideas in an efficient manner, which should be needed in order for the management to acknowledge them. The positive approach to hiring employees from other companies or the university to promote a new mindset and different perspectives is quickly demolished when the newly hired employees feel downplayed when trying to lift important ideas and findings. There seems to be a narrow-minded approach present within the organization, deriving from employees adhering to the organization for a long time.

6.4 Concluding thoughts

To me, a distinct vision of people being dissatisfied with the continuous denying and dismissal of the normative and coercive pressures emerges during the research. The employees require better, more distinct, penetration and buy-in into their ideas and initiatives by the management, to ease the frustration and stake out the continuing journey towards realizing the Evergreen strategy. Right now, the interviewees for example are demanding smarter investment strategies for IFS Applications, but they are also starting to speculate whether there are people in the hierarchy that are afraid of change or if IFS Applications is a dying cash-cow. While the Evergreen strategy is pointed out as a main strategical investment target, the 'hand' steering towards it seems to be absent when listening to the interviewees. Therefore, they all have faith in the new CEO to act. He has started his

commitment strongly by communicating loud and clear, that change is needed and that IFS has to act now.

While the enterprise systems segment in general is slow moving compared to other fast-paced segments of the software industry, it is not yet fully responding to the coercive isomorphic pressure of offering a continuous update release model or the normative isomorphic pressure of adopting DevOps related practices to accomplish and conform to it. At the same time, all employees I've met during the research are dedicated to realizing the Evergreen strategy, and bring more customer value to the table. IFS has enjoyed great success thanks to its extensive customer focus and organizational decentralization. Now, when both the organization's employees and customers are affected by the society's increasing process of globalization, IFS is punished for its previous dedication to deliver customer value. IFS is a small player in the enterprise systems market, carrying a product as competent as the ones originating from the large players. The organization must find an exit in the common release frequency problem, often mentioned as a *drama* in DevOps literature, where the difficulty of implementing a release leads to the adoption of an infrequent release schedule, which in turn worsens the implementation difficulty. Change is needed, and it is needed now.

6.5 Contribution

I argue, this research study is contributing with important insights both to the studied case company, and the theoretical fields of DevOps and Neo-institutional theory. Before the study was conducted, no DevOps-related research utilizing neo-institutional theory or Oliver's (1991) strategic response framework could be found. The study has therefore contributed with a new perspective and application for the academical field of institutionalism. Furthermore, in information systems literature, most often the definition of DevOps, and the adoption problems of certain DevOps related practices, are discussed. This study contributes with a different take on DevOps as a contextual industry trend, while exploring how a Swedish enterprise systems vendor answers to it.

The study resulted in the identification of coercive, normative and mimetic isomorphic pressures, out of which the former two are recognized as the most evident. The DevOps paradigm is identified as a combination of a coercive pressure exerted by the customers, who is demanding an efficient update release model, and a normative isomorphic pressure, exerted by the employees, who see the possibility to accommodate the customer demands and conform with the industry trends by adopting practices and routines from the DevOps concept. By emphasizing that both the institutional effects of the DevOps paradigm affecting IFS and their intra-organizational difficulties are originating from their customer focused orientation and decentralized organizational model, I hope to contribute to the understanding of similar problems arising in other parts of the software development industry. I also contribute directly to the studied organization of IFS by illuminating the employees' experienced lack of management buy-in regarding DevOps practices and idea facilitation, something they have to improve if they want to continue challenging the big players within the enterprise systems segment.

7. Reflection and future research

This journey started by utilizing an interpretive philosophical perspective together with a qualitative research model and an abductive approach. The study is of qualitative nature, and would have been impossible to perform with a quantitative model without changing the orientation of the research. The study started out with a pure inductive approach, which seemed to fit the case study method well, but thanks to my university supervisor I was influenced by the neo-institutional theory which has been helpful for setting the framing for the research. The research quality and ethics statements in the methodology chapter has both been realized to the promised levels. By focusing on triangulation throughout the whole study, for example by double-checking interesting interview topics in the IFS internal intranet, high quality could be ensured. Additionally, the empirically collected information has been reviewed by my supervisors at IFS, to further add to the research quality.

By interviewing ten employees I have based the study on a stable foundation, with a reasonable research proportion of a single researcher kept in mind. Despite interviewing various people from different divisions of the organization, with diverse perspectives and roles, I am certain that there are many more stories and perspectives to collect and analyze from the rest of the over 3500 employees within the organization of IFS. I therefore recommend IFS to further investigate the thoughts of the employees regarding the work towards Evergreen and the adoption of DevOps practices and routines as a facilitator, in order to ensure great communication and sustain good relations between all employees.

The research process has taught me many things. Firstly, that an iterative-friendly approach and attitude is desirable. Change will always occur in some way, in every project and by being open for it, and welcome it rather than denying the reasons for it, the process becomes more easy-going and productive. During this research study, the scope changed many times during the initial period, for the better. With a more positive attitude towards change, the research process would have been more positive in its whole. Secondly, discussing the research subject with many diverse people within the case company proved to be very important. The discussions often led to fruitful examples of additional interviewees or sub-topics to delve into. The lesson learned is that there is seldom something negative in discussing topics with new people with diverse backgrounds or perspectives.

I believe my background as a bachelor business student within strategy and management, together with my master studies within information systems and management has formed the study in a healthy way. To perform a study like this, great understanding for the organizational perspective as well as the software developing techniques are required. Though, I believe that further studies regarding both a more employee oriented focus, and a more technical direction could be of value, since it is there I reach my impediments. They are therefore proposed.

Within the field of research, I recommend that the DevOps paradigm is further researched from a contextual perspective. In general, within the research field, DevOps is commonly only seen as practices and routines to reach more frequent update releases. More work is needed to admit the width of the paradigm, not only covering the 'small' software developers like Spotify, but the whole spanning of it, also affecting the society and end-consumers, who gain value from organizations adopting the DevOps concept. I also propose a further study within the context of DevOps, from a Joseph Schumpeter-influenced perspective. I believe that DevOps is the next big wave of 'Creative Destruction' within technology and further insights and research regarding that convocation could be valuable.

8. References

- Avgerou, C. (2000) 'IT and organizational change: an institutional perspective', *Information Technology & People*, 13(4), 234-262.
- Barr, J. (2006) 'ACM Queue: Interview with Amazon's Werner Vogels', *AWS News Blog* 2018. [ONLINE] Available at: https://aws.amazon.com/blogs/aws/acm_queue_inter/ [Accessed 20 April 2018].
- Carey, S. & Macaulay, T. (2018) 'Windows 10: Pros and cons for enterprise', *Computer World UK*. [ONLINE] Available at: <https://www.computerworlduk.com/applications/pros-cons-for-businesses-moving-windows-10-3621064/> [Accessed 15 March 2018].
- Choudhary, V. (2007) 'Comparison of software quality under perpetual licensing and software as a service', *Journal of Management Information Systems*, 24(2), 141-165.
- Choy, L. T. (2014) 'The strengths and weaknesses of research methodology: Comparison and complimentary between qualitative and quantitative approaches', *IOSR Journal of Humanities and Social Science*, 19(4), 99-104.
- Claps, G. G., Svensson, R. B., & Aurum, A. (2015) 'On the journey to continuous deployment: Technical and social challenges along the way', *Information and Software technology*, 57, 21-31.
- Colomo-Palacios, R., García-Crespo, Á., Soto-Acosta, P., Ruano-Mayoral, M., & Jiménez-López, D. (2010) 'A case analysis of semantic technologies for R&D intermediation information management', *International Journal of Information Management*, 30(5), 465-469.
- Cockburn, I. M., & MacGarvie, M. J. (2009) 'Patents, Thickets and the Financing of Early-Stage Firms: Evidence from the Software Industry', *Journal of Economics & Management Strategy*, 18(3), 729-773.
- Dubey, A., & Wagle, D. (2007) 'Delivering software as a service', *The McKinsey Quarterly*, 6(2007), 2007.
- Dubois, A., & Gadde, L. E. (2002) 'Systematic combining: an abductive approach to case research', *Journal of business research*, 55(7), 553-560.
- Dyck, A., Penners, R., & Lichter, H. (2015) 'Towards definitions for release engineering and devops', In *Release Engineering*, 2015 IEEE/ACM 3rd International Workshop on (pp. 3-3). IEEE.
- Eisenhardt, K. M. (1989) 'Building theories from case study research', *Academy of management review*, 14(4), 532-550.
- Eisenhardt, K. M. (2002) 'Has strategy changed?' *MIT Sloan Management Review*, 43(2), 88.
- El-Gazzar, R., & Wahid, F. (2013) 'An Analytical Framework to Understand the Adoption of Cloud Computing: An Institutional Theory Perspective', *Proceedings of the International Conference on Cloud Security Management: ICCSM 2013* (p. 91). Academic Conferences Limited.
- El-Gazzar, R., & Wahid, F. (2015) 'Strategies for Cloud Computing Adoption: Insights from the Norwegian Public Sector', In *Proceedings of the 12 European, Mediterranean & Middle Eastern Conference on Information Systems (EMCIS 2015)*.

Farroha, B. S., & Farroha, D. L. (2014) 'A framework for managing mission needs, compliance, and trust in the DevOps environment' *In Military Communications Conference (MILCOM)*, 2014 IEEE (pp. 288-293). IEEE.

Fish, J. (2018) 'The Practical Guide to Enterprise DevOps and Continuous Delivery', *Micro Focus*. [ONLINE] Available at: <https://www.microfocus.com/media/ebook/Software-DevOps-eBook.pdf> [Accessed 20 April 2018].

Fitzgerald, B., & Stol, K. J. (2014) 'Continuous software engineering and beyond: trends and challenges', *In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, (pp. 1-9). ACM.

Fowler, M. (2018) 'Frequency Reduces Difficulty', *martinfowler.com* [ONLINE] Available at: <https://martinfowler.com/bliki/FrequencyReducesDifficulty.html> [Accessed 19 April 2018].

Greenberg, E., Hirt, M., & Smit, S. (2017) 'The global forces inspiring a new narrative of progress' *McKinsey Quarterly*, 2, 35-52.

Humble, J., & Molesky, J. (2011) 'Why enterprises must adopt devops to enable continuous delivery', *Cutter IT Journal*, 24(8), 6.

Hu, Q., Hart, P., & Cooke, D. (2007), 'The role of external and internal influences on information systems security - a neo-institutional perspective', *The Journal of Strategic Information Systems*, 16(2), 153-172.

Jansen, S., & Brinkkemper, S. (2006) 'Ten misconceptions about product software release management explained using update cost/value functions' *In Software Product Management*, 2006. IWSPM'06. International Workshop on (pp. 44-50). IEEE.

Jain, N., Mali, S. G., & Kulkarni, S. (2016) Infield firmware update: Challenges and solutions. In Communication and Signal Processing (ICCSP), 2016 International Conference on (pp. 1232-1236). IEEE.

Johnston, M. (2013) 'Mimetic, coercive and normative influences and the decision of national sport organizations to bid for world championship events', *Doctoral dissertation*, (Auckland University of Technology).

Justesen, L., & Mik-Meyer, N. (2011) *Kvalitativa metoder: Från vetenskapsteori till praktik*. Studentlitteratur, Lund.

Kamuto, M. B., & Langerman, J. J. (2017) 'Factors inhibiting the adoption of DevOps in large organisations: South African context' *In Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017 2nd IEEE International Conference on (pp. 48-51). IEEE.

Kaplan, B., & Maxwell, J. A. (2005) 'Qualitative research methods for evaluating computer information systems', *In Evaluating the organizational impact of healthcare information systems* (pp. 30-55). Springer, New York, NY.

Katkalo, V. S., Pitelis, C. N., & Teece, D. J. (2010) 'Introduction: On the nature and scope of dynamic capabilities', *Industrial and Corporate Change*, 19(4), 1175-1186.

- Khoo, H. M., & Robey, D. (2007) 'Deciding to upgrade packaged software: a comparative case study of motives, contingencies and dependencies', *European Journal of Information Systems*, 16(5), 555-567.
- Lahtela, A., & Jäntti, M. (2011) 'Challenges and problems in release management process: A case study', *In Software Engineering and Service Science (ICSESS)*, 2011 IEEE 2nd International Conference on (pp. 10-13). IEEE.
- Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2016) 'Relationship of DevOps to Agile, Lean and Continuous Deployment', *In International Conference on Product-Focused Software Process Improvement* (pp. 399-415). Springer, Cham.
- Mays, N., & Pope, C. (2000) 'Assessing quality in qualitative research', *Bmj*, 320(7226), 50-52.
- Mockus, A., & Weiss, D. (2008) 'Interval quality: Relating customer-perceived quality to process quality', *In Proceedings of the 30th international conference on Software engineering* (pp. 723-732). ACM.
- Microsoft (2018) 'Overview of Windows as a service (Windows 10)' *Microsoft Docs*. [ONLINE] Available at: <https://docs.microsoft.com/en-us/windows/deployment/update/waas-overview> [Accessed 15 March 2018].
- Mignerat, M., & Rivard, S. (2009) 'Positioning the institutional perspective in information systems research', *Journal of Information Technology*, 24(4), 369-391.
- Mills, A. (2017) 'Here's How Long the iOS 11.0.3 Update Takes' *GottaBe Mobile*. [ONLINE] Available at: <https://www.gottabemobile.com/how-long-does-the-ios-11-update-take/> [Accessed 14 February 2018].
- Myers, M. D. (1997) 'Qualitative research in information systems', *Management Information Systems Quarterly*, 21(2), 241-242.
- Myers, M. D. (2013). *Qualitative research in business and management*. Sage, Pittsburgh.
- Neely, S., & Stolt, S. (2013) 'Continuous delivery? easy! just change everything (well, maybe it is not that easy)' *In Agile Conference (AGILE)*, 2013 (pp. 121-128). IEEE.
- Oliver, C. (1991) 'Strategic responses to institutional processes' *Academy of management review*, 16(1), 145-179.
- Olsson, H. H., Alahyari, H., & Bosch, J. (2012) 'Climbing the "Stairway to Heaven"--A Multiple-Case Study Exploring Barriers in the Transition from Agile Development towards Continuous Deployment of Software', *In Software Engineering and Advanced Applications (SEAA)*, 2012 38th EUROMICRO Conference on (pp. 392-399). IEEE.
- Porter, M. E., & Heppelmann, J. E. (2014) 'How smart, connected products are transforming competition' *Harvard Business Review*, 92(11), 64-88.
- Ray, S., Jin, Y., & Raychowdhury, A. (2016) 'The changing computing paradigm with internet of things: A tutorial introduction', *IEEE Design & Test*, 33(2), 76-96.

- Shahin, M., Babar, M. A., Zahedi, M., & Zhu, L. (2017) 'Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges', *In Empirical Software Engineering and Measurement (ESEM)*, 2017 ACM/IEEE International Symposium on (pp. 111-120). IEEE.
- Smeds, J., Nybom, K., & Porres, I. (2015). 'DevOps: a definition and perceived adoption impediments', *In International Conference on Agile Software Development* (pp. 166-177). Springer, Cham.
- Suchman, M. C. (1995) 'Managing legitimacy: Strategic and institutional approaches', *Academy of management review*, 20(3), 571-610.
- Suddaby, R. (2010) 'Challenges for institutional theory', *Journal of management inquiry*, 19(1), 14-20.
- Su, N. (2011) 'Emergence of cloud computing: an institutional innovation perspective', *Thirty Second International Conference on Information Systems*, Shanghai 2011.
- Vetenskapsrådet (2017) 'Research ethics', *Vetenskapsrådet – The Swedish Research Council*. [ONLINE] Available at: <https://www.vr.se/inenglish/ethics.4.69f66a93108e85f68d48000116.html> [Accessed 19 February 2018].
- Teece, D. J., Pisano, G., & Shuen, A. (1999) 'Dynamic capabilities and strategic management', *In Knowledge and strategy* (pp. 77-115).
- Turner, M., Budgen, D., & Brereton, P. (2008) 'Turning software into a service', *Computer*, 36(10), 38-44.
- Van der Hoek, A., & Wolf, A. L. (2003) 'Software release management for component-based software', *Software: Practice and Experience*, 33(1), 77-98.
- Vania, K., & Rashidi, Y. (2016) 'Tales of software updates: The process of updating software', *In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 3215-3226). ACM.
- Wettinger, J., Breitenbücher, U., & Leymann, F. (2014) 'DevOpSlang—bridging the gap between development and operations', *In European Conference on Service-Oriented and Cloud Computing* (pp. 108-122). Springer, Berlin, Heidelberg.
- Xin, M., & Levina, N. (2008) 'Software-as-a-service model: Elaborating client-side adoption factors', *Twenty Ninth International Conference on Information Systems*, Paris 2008
- Zhang, J., & Seidmann, A. (2010) 'Perpetual versus subscription licensing under quality uncertainty and network externality effects', *Journal of management information systems*, 27(1), 39-68.